Chin. J. Chem. Eng., **15**(4) 533—538 (2007)

Modeling and Control of a Continuous Stirred Tank Reactor Based on a Mixed Logical Dynamical Model*

DU Jingjing(杜静静), SONG Chunyue(宋春跃) and LI Ping(李平)**
State Key Lab of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China

Abstract A novel control strategy for a continuous stirred tank reactor (CSTR) system, which has the typical characteristic of strongly pronounced nonlinearity, multiple operating points, and a wide operating range, is initiated from the point of hybrid systems. The proposed scheme makes full use of the modeling power of mixed logical dynamical (MLD) systems to describe the highly nonlinear dynamics and multiple operating points in a unified framework as a hybrid system, and takes advantage of the good control quality of model predictive control (MPC) to design a controller. Thus, this approach avoids oscillation during switching between sub-systems, helps to relieve shaking in transition, and augments the stability robustness of the whole system, and finally achieves optimal (*i.e.* fast and smooth) transition between operating points. The simulation results demonstrate that the presented approach has a satisfactory performance.
Keywords continuous stirred tank reactor, mixed logical dynamical model, multiple-operating point, state transition, hybrid system

1 INTRODUCTION

Continuous stirred tank reactors (CSTRs) are common chemical devices and also important technological sectors of the chemical process industry, which exhibit highly nonlinear behaviors and usually have wide operating ranges. In addition, nowadays, CSTRs often have to operate in multiple operating regions to manufacture several different products to realize flexible manufacture, and enhance competition ability. Hence, a very important control objective is to minimize the product transition time, and thereby reduce the amount of off-specification product produced during transition in such situation[1]. However, the nonlinear behavior of CSTRs becomes more significant during these product transitions as compared to local operation around a steady state. As a result, CSTR control provides unique opportunities for employing novel transition control technique.

In the past decade, considerable research efforts have been made for the modeling and control of nonlinear systems, and quite a few control strategies have been proposed, such as linearization at a local steady state[2], exact feedback linearization[3], gain-scheduling[3], multi-model self-adaptive control (MMAC)[4], multi-model control[1,2], and so on. Linearization at a local steady state requires a relatively stable operating point, and is not appropriate for widely nonlinear systems. The exact feedback linearization calls for an exact model of the controlled system and all the states of the system to be measurable, which are often impossible in practice. Gain-scheduling approximates a nonlinear system by a series of time-invariant linear sub-systems around a group of given operating points. Control performances cannot be guaranteed if the system strays off those set points. As for multi-model adaptive control, it is difficult to decide the number of models that can span the whole operation region. Multi-model control has a hidden disadvantage of oscillating during switching between linear sub-systems.

This article studies the modeling and control of a CSTR system based on the hybrid system theory. A mixed logical dynamical (MLD) model based model predictive control (MPC) approach is proposed for the CSTR system with high nonlinearity, multiple operating points, and a wide operating range. The CSTR system explained in this article is a nonlinear system that can be approximated by piecewise linear functions. The CSTR is modeled as a hybrid system in the MLD form for three main reasons. First, the MLD formulation captures the wide nonlinearity of the CSTR system, and describes the hybrid feature of multiple operating points of the CSTR system in a general framework. Second, it allows defining the optimal control problem in a convenient way. Third, an effective HYSDEL software package is available to obtain an MLD model conveniently[5], and a piecewise affine (PWA) system can be converted to an MLD representation directly and conveniently.

2 CONTINUOUS STIRRED TANK REACTOR SYSTEM

Consider a standard two-state CSTR with an exothermic irreversible first-order reaction A→B, as shown in Fig.1. C_A is the concentration of resultant A; T is the temperature of the reactor; q_c is the flow rate of the coolant; T_{cf} is the temperature of the coolant. C_A is the output of the CSTR system, and T_{cf} is the input.

The dynamics of the system can be described by the following nonlinear equations[1]:

$$\begin{cases} \dfrac{dx_1}{dt} = -\phi x_1 \kappa(x_2) + q(x_{1f} - x_1) \\ \dfrac{dx_2}{dt} = \beta\phi x_1 \kappa(x_2) - (q+\delta)x_2 + \delta u + q x_{2f} \\ y = x_1 \end{cases} \tag{1}$$

Received 2006-09-16, accepted 2007-03-14.
* Supported by the National Natural Science Foundation of China (No.60404018) and the State Key Development Program for Basic Research of China (No.2002CB312200).
** To whom correspondence should be addressed. E-mail: csong@umd.edu

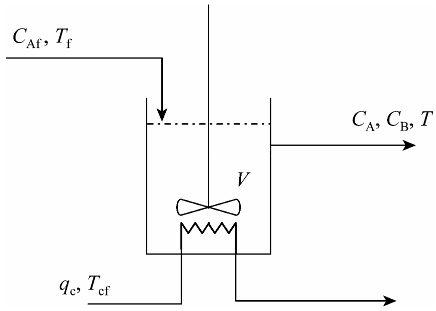


Figure 1 Continuous stirred tank reactor

where, x_1 is the resultant concentration C_A, x_2 is the reactor temperature T, and u is the coolant temperature T_{cf}. Let $\boldsymbol{x}=[x_1, x_2]^T$, the other parameters are as follows:

$$\kappa(x_2) = \exp\left(\frac{x_2}{1+x_2/\lambda}\right), \phi = 0.072, q = 1.0, \beta = 8.0,$$
$$\delta = 0.3, \lambda = 20.0, x_{1f} = 1.0, x_{2f} = 0.$$

Under the nominal operating conditions, say $u=0$, the reactor exhibits three steady states (operating points):

$$\textbf{xs1}: (x_1, x_2) = (0.856, 0.886);$$
$$\textbf{xs2}: (x_1, x_2) = (0.5528, 2.7517);$$
$$\textbf{xs3}: (x_1, x_2) = (0.2353, 4.7050).$$

And the ranges of the variables are as follows:

$$\boldsymbol{x} \in [0,1]\times[0,6]; u \in [-2,2]; R:[0,1]\times[0,6].$$

3 MLD SYSTEMS AND MPC TECHNIQUE

3.1 MLD systems

MLD systems are described by interacting physical laws, logic rules, and operating constraints in linear dynamic equations subject to linear mixed-integer inequalities. Proposition logic is used to represent logic rules, heuristic knowledge, and logic constraints in the system, and is then transformed into linear inequalities. Detailed transforming rules and the set-ups of MLD formulation can be found in Ref.[6]. Mixed logical systems are a versatile framework to model various classes of systems, among which there are linear hybrid systems, finite state machines, some classes of discrete event systems, constrained linear systems, and nonlinear systems whose nonlinearities can be expressed or suitably approximated by piecewise linear functions[6].

The general form of MLD hybrid systems as introduced in Ref.[6] is:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}_1\boldsymbol{u}(k) + \boldsymbol{B}_2\boldsymbol{\delta}(k) + \boldsymbol{B}_3\boldsymbol{z}(k) \tag{2a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}_1\boldsymbol{u}(k) + \boldsymbol{D}_2\boldsymbol{\delta}(k) + \boldsymbol{D}_3\boldsymbol{z}(k) \tag{2b}$$

$$\boldsymbol{E}_2\boldsymbol{\delta}(k) + \boldsymbol{E}_3\boldsymbol{z}(k) \leqslant \boldsymbol{E}_1\boldsymbol{u}(k) + \boldsymbol{E}_4\boldsymbol{x}(k) + \boldsymbol{E}_5 \tag{2c}$$

where, $k \in \mathbb{Z}$, $\boldsymbol{x} \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ denotes the states of the system, $\boldsymbol{u} \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ denotes the inputs and $\boldsymbol{y} \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$ denotes the outputs, with both real and binary components. Furthermore, $\boldsymbol{\delta} \in \{0,1\}^{\gamma_l}$ and $\boldsymbol{z} \in \mathbb{R}^{\gamma_c}$ represent the binary and auxiliary continuous variables, respectively. All constraints including both operating constraints and those transformed from propositional logic are summarized in the inequality (2c). Note that although the description (2c) seems to be linear, nonlinearity is hidden in the integrality constraints over the binary variables.

3.2 MPC based on MLD model

The main idea of MPC is to use a model of the system to predict the future evolution of the system in a fixed prediction horizon with the measurements of the system. Based on this prediction at each time step k, the controller selects a sequence of future command inputs through an optimization procedure, which aims at minimizing a suitable cost function and enforces fulfillment of the constraints. Then, only the first sample of the optimal sequence is applied to the plant at time step k, and at time step $k+1$, the whole optimization procedure is repeated with new plant measurements. This online replanning provides the desired feedback control action.

Let k be the current time step, N is the prediction horizon, and $\boldsymbol{x}(k)$ is the current state. $\boldsymbol{x}(i\,|\,k)$ is the state predicted at time step $k+i$ according to $\boldsymbol{x}(k)$ and input sequence $\boldsymbol{u}_k^{N-1} \triangleq \{\boldsymbol{u}(k), \boldsymbol{u}(k+1), \cdots, \boldsymbol{u}(k+N-1)\}$, and $\boldsymbol{\delta}(i\,|\,k)$, $\boldsymbol{z}(i\,|\,k)$, $\boldsymbol{y}(i\,|\,k)$ are similarly defined, $i=1,\cdots,N$. Consider the following optimal control problem[6]:

$$\min_{\{u(k+i), i=0,1,\cdots,n-1\}} J\left[\boldsymbol{u}_k^{N-1}, \boldsymbol{x}(k)\right] \triangleq \sum_{i=0}^{N-1}\Big(\left\|\boldsymbol{u}(i)-u_e\right\|_{\boldsymbol{Q}_1}^P + \left\|\boldsymbol{\delta}(i\,|\,k)-\delta_e\right\|_{\boldsymbol{Q}_2}^P + \left\|\boldsymbol{z}(i\,|\,k)-z_e\right\|_{\boldsymbol{Q}_3}^P + \left\|\boldsymbol{x}(i\,|\,k)-x_e\right\|_{\boldsymbol{Q}_4}^P + \left\|\boldsymbol{y}(i\,|\,k)-y_e\right\|_{\boldsymbol{Q}_5}^P\Big) \tag{3a}$$

s. t.

$$\begin{cases} \boldsymbol{x}(N\,|\,k) = \boldsymbol{x}_e \\ \boldsymbol{x}(i+1\,|\,k) = \boldsymbol{A}\boldsymbol{x}(i\,|\,k) + \boldsymbol{B}_1\boldsymbol{u}(i) + \boldsymbol{B}_2\boldsymbol{\delta}(i\,|\,k) + \boldsymbol{B}_3\boldsymbol{z}(i\,|\,k) \\ \boldsymbol{y}(i\,|\,k) = \boldsymbol{C}\boldsymbol{x}(i\,|\,k) + \boldsymbol{D}_1\boldsymbol{u}(i) + \boldsymbol{D}_2\boldsymbol{\delta}(i\,|\,k) + \boldsymbol{D}_3\boldsymbol{z}(i\,|\,k) \\ \boldsymbol{E}_2\boldsymbol{\delta}(i\,|\,k) + \boldsymbol{E}_3\boldsymbol{z}(i\,|\,k) \leqslant \boldsymbol{E}_1\boldsymbol{u}(i) + \boldsymbol{E}_4\boldsymbol{x}(i\,|\,k) + \boldsymbol{E}_5 \end{cases} \tag{3b}$$

where, $\boldsymbol{Q}_j = \boldsymbol{Q}_j' \geqslant 0$, $j=1,\cdots,5$ are the weighted matrix, and $\boldsymbol{x}_e, \boldsymbol{u}_e, \boldsymbol{\delta}_e, \boldsymbol{z}_e, \boldsymbol{y}_e$ are the values of the steady point that satisfy Eq.(2).

It is known that the control sequence solved from Eqs.(3a), (3b) can guarantee the MLD system (2) stability. The stability proof is detailed in Ref.[6].

This optimal problem can be cast as a mixed integer programming problem, which is solved online.

When $P=1$ or ∞, the problem is turned into a mixed integer linear programming (MILP) problem, and when $P=2$, it is turned into a mixed integer quadratic programming (MIQP) problem. By solving the MILP or MIQP problem at each time step k, the optimal control sequence $\boldsymbol{u}(k)$, $\boldsymbol{u}(k+1)$, ⋯, $\boldsymbol{u}(k+N-1)$ can be computed. According to the moving horizon philosophy, only the first sample $\boldsymbol{u}(k)$ is applied to the system, and all the rest are disregarded. At the next time step $k+1$, the whole procedure is repeated.

Suppose $P=2$ in the performance index J in Eq.(3a); the MIQP problems will have to be solved online to obtain the control law. When compared to MILP problems, however, MIQP problems indicate onerous and slow computation, and troublesome parameter tuning. On the other hand, it is more natural to express the performance indexes of the practical control problems in linear objective functions[7]. Therefore, ∞-norm linear performance index is chosen as our objective function of the CSTR system.

The earlier proposed multi-model control strategy for nonlinear systems has hidden disadvantages. It is difficult to schedule the linear subsystems coordinately and may incur oscillation while switching between different subsystems. However, the MLD model introduced in this article covers all the linear subsystems in a unified framework, which helps to relieve the oscillation during switching, and augments the stability robustness of the whole system[7].

4 THE MLD MODEL OF THE CSTR SYSTEM

4.1 Linearization

The full state space R:[0,1]×[0,6] is divided into three sub-regions R_1:[0.78,1]×[0,6]; R_2:[0.35,0.78]×[0,6]; R_3:[0,0.35]×[0,6], and each has a steady point in it. By linearizing the nonlinear system (1) around each steady point in each sub-region and discretizing it with sampling period $T_s=0.1$s, the PWA formulation of the CSTR system is as follows:

$$\begin{cases} \boldsymbol{x}(k+1)=\begin{cases} \boldsymbol{A}_{11}\boldsymbol{x}(k)+\boldsymbol{B}_{11}\boldsymbol{u}(k)+\boldsymbol{b}_{11}, \boldsymbol{x}\in R_1 \\ \boldsymbol{A}_{22}\boldsymbol{x}(k)+\boldsymbol{B}_{22}\boldsymbol{u}(k)+\boldsymbol{b}_{22}, \boldsymbol{x}\in R_2 \\ \boldsymbol{A}_{33}\boldsymbol{x}(k)+\boldsymbol{B}_{33}\boldsymbol{u}(k)+\boldsymbol{b}_{33}, \boldsymbol{x}\in R_3 \end{cases} \\ \boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k) \end{cases} \tag{4}$$

where,

$$\begin{cases} \boldsymbol{A}_{11}=\begin{bmatrix} 0.8889 & -0.0123 \\ 0.1254 & 0.9751 \end{bmatrix}, \boldsymbol{B}_{11}=\begin{bmatrix} -0.0002 \\ 0.0296 \end{bmatrix}, \boldsymbol{b}_{11}=\begin{bmatrix} 0.1060 \\ -0.0852 \end{bmatrix} \\ \boldsymbol{A}_{22}=\begin{bmatrix} 0.8241 & -0.0340 \\ 0.6365 & 1.1460 \end{bmatrix}, \boldsymbol{B}_{22}=\begin{bmatrix} -0.0005 \\ 0.0322 \end{bmatrix}, \boldsymbol{b}_{22}=\begin{bmatrix} 0.1907 \\ -0.7537 \end{bmatrix}, \boldsymbol{C}=[1\ \ 0]. \\ \boldsymbol{A}_{33}=\begin{bmatrix} 0.6002 & -0.0463 \\ 2.4016 & 1.2430 \end{bmatrix}, \boldsymbol{B}_{33}=\begin{bmatrix} -0.0007 \\ 0.0338 \end{bmatrix}, \boldsymbol{b}_{33}=\begin{bmatrix} 0.3119 \\ -1.7083 \end{bmatrix} \end{cases}$$

$k\in\mathbb{Z}$ it is the short form of kT_s.

Obviously, $R_1\cap R_2=\varnothing$, $R_1\cap R_3=\varnothing$, $R_2\cap R_3=\varnothing$, and $R_1\cup R_2\cup R_3=R$. System (4) covers the whole operating space of system (1), and it is a proper approximation of system (1). According to the definition and expounding of well-posedness in Ref.[6], system (4) is completely well-posed. In fact, a model derived from a real system is well posed. Well-posed and bounded PWA systems can be rewritten as MLD systems[6].

4.2 Transforming into MLD form

Since system (4) is completely well-posed, it can be cast as an MLD model. Two logic variables δ_1 and δ_2 are introduced, and defined as:

$$\begin{cases} \delta_1=1\leftrightarrow[1\ \ 0]\boldsymbol{x}\leqslant 0.35 \\ \delta_2=1\leftrightarrow[1\ \ 0]\boldsymbol{x}\leqslant 0.78 \end{cases}, \text{ then } \delta_1=1\rightarrow\delta_2=1 \tag{5}$$

Let $\boldsymbol{d}=\left[\delta_1(k),\delta_2(k)\right]$; then, when the system works in region R_1, the logic vector $\boldsymbol{d}=[0,0]$; when in R_2, $\boldsymbol{d}=[0,1]$, and when in R_3, $\boldsymbol{d}=[1,1]$.

Thus, Eq.(4) can be rewritten as:

$$\begin{cases} \boldsymbol{x}(k+1)=\left[\boldsymbol{A}_{11}\boldsymbol{x}(k)+\boldsymbol{B}_{11}\boldsymbol{u}(k)+\boldsymbol{b}_{11}\right](1-\delta_1)+ \\ \qquad \left[\boldsymbol{A}_{22}\boldsymbol{x}(k)+\boldsymbol{B}_{22}\boldsymbol{u}(k)+\boldsymbol{b}_{22}\right](1-\delta_1)\delta_2+ \\ \qquad \left[\boldsymbol{A}_{33}\boldsymbol{x}(k)+\boldsymbol{B}_{33}\boldsymbol{u}(k)+\boldsymbol{b}_{33}\right]\delta_1 \\ \boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k) \end{cases} \tag{6}$$

Introduce 6 auxiliary variables $z_1(k)$, $z_2(k)$, $z_3(k)$, $z_4(k)$, $z_5(k)$, $z_6(k)$ and define them as:

$$\begin{cases} \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix}=\left[\boldsymbol{A}_{11}\boldsymbol{x}(k)+\boldsymbol{B}_{11}\boldsymbol{u}(k)+\boldsymbol{b}_{11}\right](1-\delta_1) \\ \begin{bmatrix} z_3(k) \\ z_4(k) \end{bmatrix}=\left[\boldsymbol{A}_{22}\boldsymbol{x}(k)+\boldsymbol{B}_{22}\boldsymbol{u}(k)+\boldsymbol{b}_{22}\right](1-\delta_1)\delta_2 \\ \begin{bmatrix} z_5(k) \\ z_6(k) \end{bmatrix}=\left[\boldsymbol{A}_{33}\boldsymbol{x}(k)+\boldsymbol{B}_{33}\boldsymbol{u}(k)+\boldsymbol{b}_{33}\right]\delta_1 \end{cases} \tag{7}$$

The operating constraints are as follows:

$$[0\ \ 0]^{\mathrm{T}}\leqslant\boldsymbol{x}\leqslant[1\ \ 6]^{\mathrm{T}} \tag{8}$$

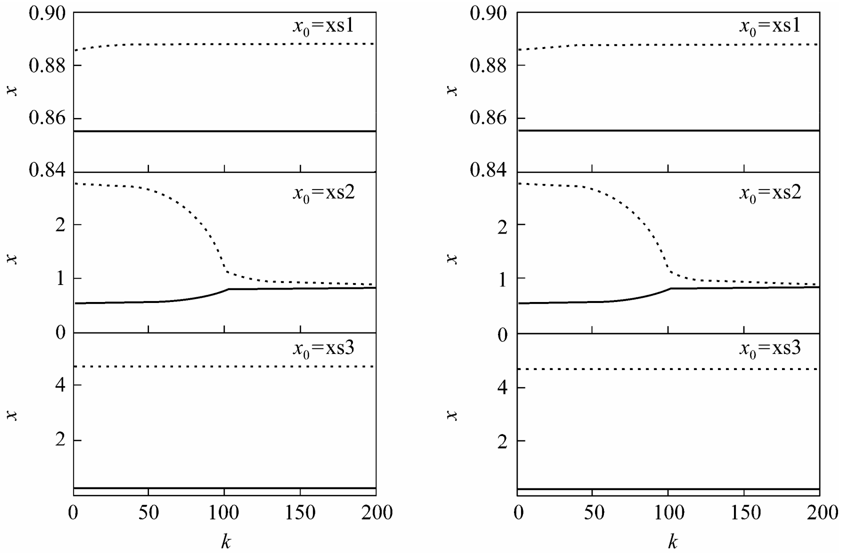
$$-2\leqslant u\leqslant 2 \tag{9}$$

Transform Eqs.(5), (7)—(9) into inequalities according to the rules in Ref.[6]. Let $\boldsymbol{\delta}(k)=\left[\delta_1(k),\delta_2(k)\right]^{\mathrm{T}}$, $\boldsymbol{z}(k)=\left[z_1(k),z_2(k),z_3(k),z_4(k),z_5(k),z_6(k)\right]^{\mathrm{T}}$. Based on these inequalities, Eq.(6) and Eq.(4) can be rewritten as:

$$\begin{cases} \boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}_1\boldsymbol{u}(k)+\boldsymbol{B}_2\boldsymbol{\delta}(k)+\boldsymbol{B}_3\boldsymbol{z}(k) \\ \boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k) \\ \boldsymbol{E}_2\boldsymbol{\delta}(k)+\boldsymbol{E}_3\boldsymbol{z}(k)\leqslant\boldsymbol{E}_1\boldsymbol{u}(k)+\boldsymbol{E}_4\boldsymbol{x}(k)+\boldsymbol{E}_5 \end{cases} \tag{10}$$

where, $\boldsymbol{A}=0$, $\boldsymbol{B}_1=0$, $\boldsymbol{B}_2=0$, $\boldsymbol{B}_3=\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$, $\boldsymbol{C}=[1\ \ 0]$. $\boldsymbol{E}_i(i=1,\cdots,5)$ of Eq.(10) is omitted due to lack of space. There are 35 inequalities in total including operating constraints and inequalities transformed from propositional logic. Eq.(10) is the MLD model of the CSTR system described in Eq.(1).

The above procedure of obtaining an MLD form of Eq.(10) can be automatized, since there exists an effective software package HYSDEL-Hybrid System Description Language[5].



(a) Evolution of the MLD model (b) Evolution of the PWA model

Figure 2 Evolution of the MLD and PWA models, starting from three operating points respectively with $u \equiv 0$

—— x_1; - - - -x_2

Since the PWA system (4) is completely well-posed, the MLD system (10) derived from it is also completely well-posed[6]. Namely, once $\boldsymbol{x}(k)$ and $\boldsymbol{y}(k)$ are assigned, the values of $\boldsymbol{\delta}(k)$ and $\boldsymbol{z}(k)$ are uniquely determined by the inequality (2c)[6]. Furthermore, $\boldsymbol{x}(k+1)$ and $\boldsymbol{y}(k)$ are uniquely solved by (2a) and (2b). This property that is usually verified while representing real plants in the MLD form, plays an important role in simulations[5]. Indeed, for completely well-posed systems, given the pair $[\boldsymbol{x}(k), \boldsymbol{u}(k)]$, it is possible to verify it by determining the values of $\boldsymbol{\delta}(k)$ and $\boldsymbol{z}(k)$ by solving a Mixed-Integer Feasibility Test. This test can be done very efficiently using the Branch and Bound algorithm[5].

5 SIMULATIONS AND RESULT ANALYSES

Under the condition $u(k) \equiv 0$, the CSTR system starts from each of the three steady points. The simulation results of the MLD and PWA models are in Fig.2. (The horizontal coordinate represents the number of sampling period; $k=20$ represents the 20th sampling period).

By comparing the subplots on the left with those on the right, it can be seen that the trajectories are entirely the same. It is confirmed that the MLD model (10) equals the PWA model (4). When the starting point is operating point xs1 or xs3, the states stay around the operating point, and remain steady. However, the system switches from sub-region R_2 to R_1, when it evolves from operating point xs2. Thus, the conclusions can be drawn that the operating points xs1 and xs3 are widely asymptotically stable, while xs2 is locally stable.

Now, the control strategy presented in Section 3 is adopted to control the CSTR system with $P=\infty$. First, consider the transition between operating points. Choose xs3 as the initial point, xs1 as the final point, and then solve the optimal problem (3) subject to Eq.(10). The resulting optimal trajectories of the CSTR system are shown in Fig.3.

Figure 3 depicts the fast and smooth transition from steady point xs3, passing by xs2, eventually to xs1. The parameters are $N=3$, $\boldsymbol{Q}_1=0$, $\boldsymbol{Q}_2=0$, $\boldsymbol{Q}_3=0.5\boldsymbol{I}_6$, $\boldsymbol{Q}_4=0.95\boldsymbol{I}_2$, $\boldsymbol{Q}_5=0.85$ ($\boldsymbol{I}_n$ is an $n\times n$ identity matrix).

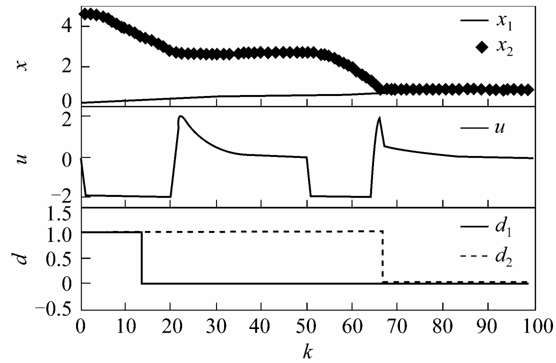


Figure 3 Evolution from xs3 by xs2 to xs1

On the whole, the transition in Fig.3 is rather fast and smooth, regardless of whether it is from sub-region R_3 to sub-region R_2 or from R_2 to R_1. The states of the system do not oscillate during the transition transient between two linear sub-systems. This is because the controller is designed and optimized in a general MLD framework, which makes the linear sub-systems well-scheduled and work coordinately. Hence, besides the stability of the system being guaranteed, the transition time is minimized, which is our important objective. The middle sub-figure of Fig.3 depicts the control sequence. Since $u(k)\in[-2, 2]$, it restricts the control input from being too large or too small, and this is why $u(k)$ in Fig.3 varies within $[-2, 2]$. Meanwhile, to achieve optimal transition, the controller will enforce the biggest command (absolute value) to the CSTR at the very beginning of every transition stage.

It is known from Eq.(4) that the control input has an opposite effect on state x_1 to x_2. That is, when u is negative, it increases x_1 but decreases x_2, yet when u is positive, it decreases x_1 but increases x_2. Meanwhile, u has a bigger force on x_2 than on x_1.

In the upper subplot of Fig.3, when the system starts from xs3=(0.2353, 4.705) at k=0 and transfers to xs2=(0.5528, 2.7517), x_1 increases from 0.2353 to 0.5528 and x_2 decreases from 4.705 to 2.7517. In the middle subfigure of Fig.3, u decreases to −2 at the very beginning, and stays unchanged in the first 20 sample intervals so as to transfer the states as quickly as possible, and reduce the transition time. To avoid overshoot and to make the states arrive at xs2 smoothly, u increases to a big positive value rapidly, and then decreases gradually to zero, when the system approaches the steady state xs2. Now, look at the state variables in the upper subplot. During the first 20 steps, the states transfer to xs2 swiftly. While getting near xs2, the states slow down, and arrive steadily. Evidently, this is consistent with the command input.

Near k=15, the system transfers from R_3 to R_2, and the logic variables change from [1, 1] to [0, 1] in the lower subplot, which agrees with the definition in Eq.(5). Since the control input has a greater force on x_2 than x_1, x_2 varies faster than x_1. x_2 reaches the steady value 2.7517 at the 20th sample period with an almost 0 static error, while x_1 reaches its steady value 0.5528 at the 30th step with a bigger static error 0.18% (although not big in fact). The transition is then complete. Thereafter, the system works steadily around the operating point xs2.

At the 50th sample period, transition from xs2 to xs1 begins, which declares that x_1 will continue to increase from 0.5528 to 0.856, while x_2 will continue to decrease from 2.7517 to 0.886. And a similar smooth and swift transition as from R_3 to R_2 recurs. Henceforth, the system remains at xs1.

From the simulation result analyses mentioned above, it can be concluded that the MPC technique based on an MLD model optimizes the control variable systematically and synthetically in a unified MLD framework and controls the system to transfer from one operating point to another quickly and smoothly, and minimizes the transition time on the premise of stability of the widely nonlinear system, without oscillating during switching from one sub-model to another.

Second, consider the transition between an operating point and a non-operating point. Fig.4 shows the evolution from $x_0=[0.8, 0.5]$ to operating point xs2. The parameters are N=3, $\boldsymbol{Q}_1$=0, $\boldsymbol{Q}_2$=0, $\boldsymbol{Q}_3=0.65\boldsymbol{I}_6$, $\boldsymbol{Q}_4=\boldsymbol{I}_2$, and $\boldsymbol{Q}_5$=0.9. It is clear that the evolution in Fig.4 is different from that in Fig.3. In this case, x_1 decreases from 0.8 to 0.5528, yet x_2 increases from 0.5 to 2.7517. In the middle subplot, u rises quickly from 0 to +2 and remains at the value of +2 in the first 38 steps to transfer the system states at its best, minimizes the time cost, and achieves optimal transition. In the upper subplot, the states vary rapidly during this time. At the 29th sample interval, the system comes into the sub-region R_1 to sub-region R_2. In the meantime, in the lower subplot, the logic vector $\boldsymbol{d}$ changes from [0, 0] to [0, 1]. After the 40th step, the system

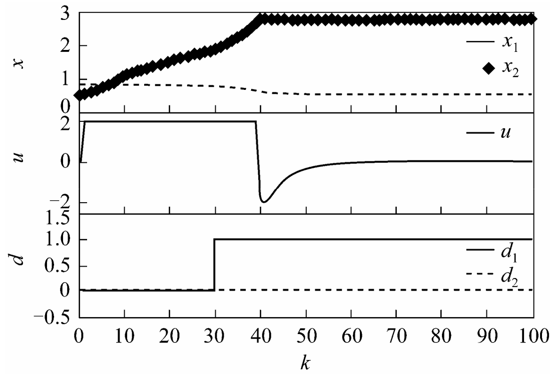


Figure 4 Transition from non-operating point x_0 to operating point xs2

approaches the steady point xs2, and meanwhile, the control input quickly falls to a big negative value from +2, and then slowly rises to zero. Thus, no overshoot happens. Thereafter, the system arrives at xs2 steadily, and it can also be found in the upper subplot that x_2 is faster than x_1 to reach the steady value with a smaller static error.

From Fig.4, it is apparent that the transition from a non-steady point to a steady point is also swift and smooth. It demonstrates again that MPC based on MLD models has a good control performance, and can implement rapid and smooth transition on the premise of the stability of widely nonlinear system.

6 CONCLUSIONS

CSTRs systems are inherently highly nonlinear, which makes the control task nontrivial. Besides, the requirement of optimal transition between operating points stimulates the need to develop flexible operating strategies. This article proposes an MLD based model predictive control approach for the CSTR systems. The MLD model, which originates from industrial practice and has a strong application background, can capture the nonlinear and hybrid feature of the CSTR system. The control law is synthesized in a unified MLD framework by a moving horizon technique, and can thus obtain optimal transition without oscillation on the premise of wide range stability of the nonlinear system as shown in the simulations.

However, there exist some drawbacks of the proposed method. For example, the MLD model of a system is not unique. If different auxiliary continuous and logic variables are introduced, a different MLD model will be obtained. Also, the biggest handicap is the computational complexity of the mixed integer programming (MIP) problems (both MIQP and MILP), which increases exponentially with the number of logic variables. More efforts are required to solve these problems.

NOMENCLATURE

C_A	the concentration of resultant A, the output of the CSTR system
C_{Af}	feed concentration of CSTR

$\boldsymbol{d}$	$\boldsymbol{d}=[\delta_1(k),\delta_2(k)]$
$\boldsymbol{I}_n$	$n\times n$ identity matrix
N	prediction horizon
n, m, p, r	dimension of $\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{\delta}$, $\boldsymbol{z}$
P	index of norm, $P=1$ refers to 2-norm, $P=$inf, refers to inf-norm
$\boldsymbol{Q}_j$	weighted matrix
q_c	the flow rate of the coolant
R	the range of the state variables
R_1, R_2, R_3	sub-regions of the whole CSTR region
T	the temperature of the reactor
T_{cf}	the temperature of the coolant, the input of the CSTR system
T_f	feed temperature
$\boldsymbol{u}(k), \cdots, \boldsymbol{u}(k+N-1)$	the optimal control sequence
$\boldsymbol{u}_k^{N-1}$	Short form of $\{\boldsymbol{u}(k),\boldsymbol{u}(k+1),\cdots,\boldsymbol{u}(k+N-1)\}$
$x_e, u_e, \delta_e, z_e, y_e$	values of the steady point
$\boldsymbol{x}(i\|k)$	state predicted at time $k+i$ according to $\boldsymbol{x}(k)$ and input sequence $\{\boldsymbol{u}_0^{N-1}\}$
xs1, xs2, xs3	steady points (operating points) of the CSTR
$z_1(k), \cdots, z_6(k)$	auxiliary continuous variables
$\boldsymbol{\delta}(i\|k), \boldsymbol{z}(i\|k), \boldsymbol{y}(i\|k)$	are similarly defined as $\boldsymbol{x}(i\|k)$
$\delta_1(k), \delta_2(k)$	auxiliary logic variables

Subscripts

c	continuous variable set
l	binary variable set

REFERENCES

1 Özkan, L., Kothare, M.V., Georgakis, C., "Control of a solution copolymerization reactor using multi-model predictive control", *Chem. Eng. Sci.*, **58**, 1207—1221 (2003).
2 Min, S., Shan, C., "Internal model control design for transition control", *AIChE J.*, **46** (2), 309—319(2000).
3 Klatt, K.U., Engell, S., "Gain-scheduling trajectory control of a continuous stirred tank reactor", *Computers Chem.Eng.*, **22**(4/5), 491—502(1998).
4 Lin, F., Yang, G., "Application of allied genetic algorithm to the CSTR system", *Control and Decision*, **16**(4), 457—464(2001).
5 Ferrari-Trecat, G., Mignone, D., Castagnoli, D., Morari, M., "Mixed logic dynamical models of a hydroelectric power plant", In: International Conference Automation of Mixed Processes Hybrid Dynamic Systems ADPM, Dortmund, Germany (2000).
6 Bemporad, A., Morari, M., "Control of systems integrating logic, dynamics, and constraints", *Automatica*, **35**(3), 407—427(1999).
7 Li, X., "Research on modeling and control of hybrid systems based on mixed logical dynamical", Ph.D. Thesis, Institute of Automation Chinese Academy of Sciences, China (2003).
8 Su, B., Chen, Z., Yuan, Z., "Multivariable decoupling predictive control with Input constraints and its application on chemical process", *Chin. J. Chem. Eng.*, **14**(2), 216—222(2006).
9 Zhang, Y., Chen, Z., Yang, P., Yuan, Z., "Multivariable nonlinear proportional-integral-derivative decoupling control based on recurrent neural net works", *Chin. J. Chem. Eng.*, **12**(5), 677—681(2004).