Research Article

A Two-Time Scale Decentralized Model Predictive Controller Based on Input and Output Model

Jian Niu, Jun Zhao, Zuhua Xu, and Jixin Qian

State Key Lab of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Zuhua Xu, zhxu@iipc.zju.edu.cn

Received 11 May 2009; Accepted 2 July 2009

A decentralized model predictive controller applicable for some systems which exhibit different dynamic characteristics in different channels was presented in this paper. These systems can be regarded as combinations of a fast model and a slow model, the response speeds of which are in two-time scale. Because most practical models used for control are obtained in the form of transfer function matrix by plant tests, a singular perturbation method was firstly used to separate the original transfer function matrix into two models in two-time scale. Then a decentralized model predictive controller was designed based on the two models derived from the original system. And the stability of the control method was proved. Simulations showed that the method was effective.

1. Introduction

Applications of singular perturbation in control theory can be traced back to 1970s [1]. When there are both fast and slow dynamics in a system, the perturbation conception can be adopted to simplify the system. Phillips first combined the optimal control theory with a singular perturbation system and presented a two-stage design of linear feedback control [2]. Later, nonlinear singular perturbation systems [3], large scale systems [4], and high-gain feedback systems [5] were investigated with optimal control theory, too. Recently, process control researchers took notice of the mechanical reasons for two-time scale characteristics of systems. And the existence of fast and slow dynamics was studied in several previous papers. Yi and Luyben analyzed the dynamic characteristic of coupled reactor/column systems in a series of papers and illuminated the mechanisms of fast and slow dynamics in some special systems [6–8]. Contou-Carrere and Daoutidis focused on the coexistence of fast and slow dynamics in integrated process networks [9]. They explained that large flow rates brought in fast dynamic and led to a time scales separation of dynamics. They also designed a precompensator for a distillation model. Sequentially, Kumar and Daoutidis further analyzed the dynamics of process with material and energy recycle, and introduced a controller design framework consisting of properly coordinated controllers in fast and slow time scales [10]. Vora and Daoutidis introduced a nonstandard form singular perturbation method and analyzed the existence of fast and slow dynamics in a nonlinear system [11]. Kumar and Christofides dealt with two-time scale chemical processes and modeled them by using nonlinear ordinary differential equations with large parameters of the form $1/\varepsilon$ [12]. They obtained a standard singularly perturbed representation. All those methods focused on process thermodynamic models for which took much work to get precise data.

In actual processes, model predictive control (MPC) was regarded as “the only advanced control methodology which has made a significant impact on industrial control engineering” [13]. And MPC has no limit of the model form. The input-output model which can be obtained easily by identification is usually used. However, all the methods mentioned above is from the point of optimal control which is an offline method. In addition, all the above methods regard the steady state output of fast dynamics as the input or known quantity of slow dynamics but pay little attention on the influence of slow dynamics on fast dynamics. MPC adopts online rolling optimization and uses feedback to correct prediction. The challenge of applying MPC in singular perturbation system becomes the compromise between the
control interval and the predictive horizon. Fast dynamic needs small control interval, while slow dynamic needs large predictive horizon. There are few papers about two-time scale MPC. Buescher and Baum introduced a two-time scale approach to nonlinear model predictive control, and used a “gapping” method to smooth the control quality [14]. But they did not consider the dynamic characteristics of the model in their control algorithm which may be not suitable for special systems.

Therefore, we focused on designing an MPC for systems with two-time scale characteristic in this paper. First, we introduced the background of this field. Then we described the two-time scale decomposition of a transfer function matrix with different dynamics in different channels. At part 3, we presented a kind of two-time scale decentralized MPC algorithm step by step and proved its stability. At last, we gave several simulations to test the validity of the two-time scale decentralized MPC algorithm.

2. Two-Time Scale Decomposition of a Transfer Function Matrix

In some systems, the dynamics varies with different channels. And the response speeds of those different channels vary so much even in different time scales. The characteristics can be got from the transfer function matrix intuitively. We simply took a two-in-two-out first-order transfer function matrix, for example and gave the following definition.

**Definition 1.** A Two-in-two-out first-order transfer function matrix \(G(s)\) with two-time scale characteristic is presented below, where

\[
G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix}
\]

\[
G_{11}(s) = \frac{a_{11}}{b_{11}s + 1} e^{\tau_{11}s}, \quad G_{12}(s) = \frac{a_{12}}{b_{12}s + 1} e^{\tau_{12}s},
\]

\[
G_{21}(s) = \frac{a_{21}}{b_{21}s + 1} e^{\tau_{21}s}, \quad G_{22}(s) = \frac{a_{22}}{b_{22}s + 1} e^{\tau_{22}s},
\]

when \(a_{11}, a_{12}, a_{21}, a_{22}\) are in the same order, \(b_{11}, b_{12} \ll b_{21}, b_{22}, \tau_{11}, \tau_{12}, \tau_{21}, \tau_{22}\) are in the same order.

Considering the system

\[
Y = G(s)U,
\]

the response speed of output \(y_1\) is much faster than that of output \(y_2\). Then a traditional central controller cannot satisfy the demands of \(y_1\) and \(y_2\) simultaneously. For example, for the fast channel a very short control interval is needed to provide enough dynamic characteristics, while for the slow channel a very large predictive horizon is needed to ensure the stability of the controller. Thus a normal model predictive controller cannot satisfy both demands. Therefore we designed a decentralized controller based on a two-time scale method.

Some papers [9, 11, 15] introduced a singular perturbation method to obtain two-time scale models which are based on state space model. In order to transform the transfer function matrix into a two-time scale form, the transfer function matrix form should be transferred into a state space form. But the delay terms cannot be expressed in a state space form. Thus we firstly took a transfer function matrix without delay to illustrate the two-time scale decomposition, and then discussed the situation with delay terms.

2.1. Without Delay Terms. Based on the method mentioned in literature[16], system

\[
G'(s) = \begin{bmatrix} G'_{11}(s) & G'_{12}(s) \\ G'_{21}(s) & G'_{22}(s) \end{bmatrix},
\]

where \(G'_{11}(s) = a_{11}/b_{11}s + 1, G'_{12}(s) = a_{12}/b_{12}s + 1, G'_{21}(s) = a_{21}/b_{21}s + 1, G'_{22}(s) = a_{22}/b_{22}s + 1, a_{11}, a_{12}, a_{21}, a_{22}\) are in the same order, \(b_{11}, b_{12} \ll b_{21}, b_{22}\), can have this form

\[
\dot{X} = AX + BU,
\]

\[
Y = CX,
\]

where

\[
A = \text{diag}[A_{11}, A_{12}, A_{21}, A_{22}],
\]

\[
B = \begin{bmatrix} B_{11} \\ B_{12} \\ B_{21} \\ B_{22} \end{bmatrix},
\]

\[
C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix};
\]

\((A_{ij}, B_{ij}, C_{ij})\) is a one-dimension state space form of \(G'_{ij}(s)\):

\[
\vdash b_{11}, b_{12} \ll b_{21}, b_{22}, |A_{11}|, |A_{12}| \gg |A_{21}|, |A_{22}|.
\]

So we can rewrite (4) in this form:

\[
\begin{bmatrix} \varepsilon \dot{x}_1 \\ \varepsilon \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \varepsilon A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \varepsilon B_{11} \\ \varepsilon B_{12} \\ B_{21} \\ B_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},
\]

\[
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix},
\]

where \(\varepsilon\) is a very small positive constant.
Journal of Automated Methods and Management in Chemistry

This form can be regarded as

\[
\begin{align*}
X &= \bar{A}_{11}X + \bar{A}_{12}Z + \bar{B}_1 U, \\
\epsilon Z &= \bar{A}_{22}X + \bar{A}_{22}Z + \bar{B}_2 U, \\
Y &= \bar{C}_1 X + \bar{C}_2 Z + \bar{D} U,
\end{align*}
\]

where

\[
\begin{align*}
\bar{A}_{11} &= \begin{bmatrix} A_{21} & 0 \\ 0 & A_{22} \end{bmatrix}, & \bar{A}_{12} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, & \bar{A}_{21} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\
\bar{A}_{22} &= \begin{bmatrix} \epsilon A_{11} & 0 \\ 0 & \epsilon A_{12} \end{bmatrix}, & \bar{B}_1 &= \begin{bmatrix} B_{21} & 0 \\ 0 & B_{22} \end{bmatrix}, \\
\bar{B}_2 &= \begin{bmatrix} \epsilon B_{11} & 0 \\ 0 & \epsilon B_{12} \end{bmatrix}, & \bar{C}_1 &= \begin{bmatrix} 0 & 0 \\ C_{21} & C_{22} \end{bmatrix}, \\
\bar{C}_2 &= \begin{bmatrix} C_{11} & C_{12} \\ 0 & 0 \end{bmatrix}, & \bar{D} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.
\end{align*}
\]

And the system in this form was discussed in several literatures [9, 15, 17], too. We got the slow model (8) and the fast model in \( \tau \)-time scale by

\[
G_f(s) = \frac{C_f(sI - A_f)^{-1}B_f + D_f}{sI - \bar{A}_{22}} + \bar{D}
\]

\[
= \frac{\epsilon a_{11}}{b_{11}s + \epsilon} + \frac{\epsilon a_{12}}{b_{12}s + \epsilon},
\]

(14)

and the fast model in \( t \)-time scale is

\[
G_r(s) = G_f(\epsilon s)
\]

\[
= \begin{bmatrix} a_{11} & a_{12} \\ b_{11}s + 1 & b_{12}s + 1 \end{bmatrix}.
\]

(15)

\( G_f(s), G_r(s), \) and \( G_t(s) \) are all descriptions from different points of the real system \( G'(s) \). And they are all two-in-two-out systems but have less state variables than the real system \( G'(s) \). If we choose a long enough sample interval for \( G'(s) \) and \( G_r(s) \), the sample values can be similar, because the fast responses turn to steady states that can be regarded as constants in a very short time. If we choose a very short sample interval for \( G'(s) \) and \( G_t(s) \) in a short enough period of time, the sample values can be similar, too. Because in such a short period of time the slow channel response is so slow that can be regarded as zero. But the output of the slow channel is not zero. Therefore, we modified the fast model in \( t \)-time scale when we designed a controller based on this fast model. Literature [18] proved the relationship between \( G_r(s) \) and \( G_f(s) \), and also the transfer function of the original system \( G'(s) \):

\[
\lim_{s \to \infty} G_r(s) = D_r = \lim_{s \to \infty} G_f(s),
\]

(16)

\[
G'(s) = G_r(s) + G_f(\epsilon s) - D_r + O(\epsilon).
\]

(17)

Equation (16) denotes that the initial value of the slow model equals to the final value of the fast model. And (17) denotes that the original system can be regarded as a sum of the slow model, the fast model, and a very little item \( O(\epsilon) \). Let

\[
O(\epsilon) = \begin{bmatrix} 0 & 0 \\ O_{21}(\epsilon) & O_{22}(\epsilon) \end{bmatrix},
\]

(18)

and let

\[
G_t(s) = G_r(s) + O(\epsilon)
\]

\[
= \begin{bmatrix} \frac{a_{11}}{b_{11}s + 1} & \frac{a_{12}}{b_{12}s + 1} \\ O_{21}(\epsilon) & O_{22}(\epsilon) \end{bmatrix} = \begin{bmatrix} G_{t1}(s) & G_{t2}(s) \\ G_{t1}(s) & G_{t2}(s) \end{bmatrix}.
\]

(19)

We had expressions of the fast model \( G_f(s) \) and the slow model \( G_t(s) \) about the model \( G'(s) \) without delay terms in \( t \)-time scale. Next we would consider the model with delay terms.
2.2. With Delay Terms. We took system (2), where \( G(s) \) shows two-time scale characteristic, as an example. Let

\[
\begin{align*}
X_{11}(s) &= G_{11}(s)U_{1}(s) = G'_{11}(s)U_{1}(s)e^{\tau_{11}s}, \\
X_{12}(s) &= G_{12}(s)U_{2}(s) = G'_{12}(s)U_{2}(s)e^{\tau_{12}s}, \\
X_{21}(s) &= G_{21}(s)U_{1}(s) = G'_{21}(s)U_{1}(s)e^{\tau_{21}s}, \\
X_{22}(s) &= G_{22}(s)U_{2}(s) = G'_{22}(s)U_{2}(s)e^{\tau_{22}s},
\end{align*}
\]

(20)

Here \( G_{ij}(s) \), \( i, j = 1, 2 \), denotes the transition process, and \( e^{\tau_{ij}s} \), \( i, j = 1, 2 \), denotes the delay time. When \( \tau_{11}, \tau_{12}, \tau_{21}, \) and \( \tau_{22} \) are in the same order, \( G_{ij}(s) \), \( i, j = 1, 2 \), reflects the main dynamic characteristic. We can get the fast model in t-time scale

\[
G_{td}(s) = \begin{bmatrix}
G'_{11}(s)e^{\tau_{11}s} & G'_{12}(s)e^{\tau_{12}s} \\
0_{21}(\epsilon) & 0_{22}(\epsilon)
\end{bmatrix}.
\]

(21)

and the slow model in t-time scale

\[
G_{sd}(s) = \begin{bmatrix}
a_{11} & a_{12} \\
G_{21}(s)e^{\tau_{21}s} & G_{22}(s)e^{\tau_{22}s}
\end{bmatrix}
\]

(22)

Then we can design a decentralized controller based on characteristics of the fast model \( G_{td}(s)(G_{td}(s)) \) and the slow model \( G_{sd}(s)(G_{sd}(s)) \).

3. Two-Time Scale Decentralized MPC

MPC is the only advanced control methodology which has made a significant impact on industrial control engineering [13]. And MPC is based on a predicted model. The MPC algorithm can be regarded as a combination of three parts: model prediction, roll optimization, and feedback rectification [19].

For the two-in-two-out system mentioned above with two-time scale characteristic, we designed a decentralized controller based on different time scales. We took the model without delay \( (G'(s)) \) to illustrate the algorithm. The fast model provided abundant fast dynamic information to ensure the control quality. And the control interval was determined by the fast model. The slow model provided prediction horizon long enough to ensure the controller’s stability. In order to illustrate the algorithm, we defined that \( P_{i} \) is the prediction horizon, \( T_{i} \) is the sampling interval based on the slow model, \( P_{j} \) is the prediction horizon, \( T_{j} \) is the sampling interval based on fast model, and \( M \) is the manipulate horizon. \( a_{ij}(t) \) is the step response of \( y_{i} \) from \( u_{j} \) at \( T_{j} \) sample interval. We got the model vector \( a_{ij} = [a_{i1} \cdots a_{ij(N)}]^T, i = 1, 2, j = 1, 2, \) and \( N \) is a number large enough to fully reflect the fast and the slow part of the model. The two-time scale decentralized DMC algorithm was shown as follows.

Step 1. Model prediction based on slow model \( G_{sd}(s) \).

On the t-time scale, the fast dynamic achieved a steady state. The model can be fully expressed by information not so necessary as \( a_{ij} \). Let \( a_{ij} = [a_{i1} \cdots a_{ij(P_{j})}]^T \), \( j = 1, 2 \), be the slow model vector, where \( a_{ij}(i) = [a_{i1}(i(1)*T_{j})+1 \cdots a_{i1}(i(T_{j})+1)] \), \( i = 1, \ldots, P_{i} \). The slow predict model is

\[
\tilde{y}_{i,PM}(p) = \tilde{y}_{i,P0}(p) + A_{i}\Delta u, \quad j = 1, 2,
\]

(23)

where, \( \tilde{y}_{i,PM}(p) \in R^{p} \) is the predict value of future output and \( \tilde{y}_{i,P0}(p) \in R^{p} \) is the prime predict value of future output:

\[
A_{i} = \begin{bmatrix}
a_{i1}(1) & 0 \\
\vdots & \ddots \\
a_{i1}(P_{i}) & \cdots & a_{ij}(P_{j}) (P_{j} - M + 1)
\end{bmatrix}
\]

(24)

Step 2. Feedback Correction based on slow model \( G_{sd}(s) \).

Let error vector be

\[
e_{i}(p + 1) = y_{i}(p + 1) - \tilde{y}_{i,PM}(p + 1),
\]

(25)

and we can get

\[
\tilde{y}_{i,cor}(p + 1) = \tilde{y}_{i,PM}(p) + H_{i}e_{i}(p + 1),
\]

(26)

where

\[
H_{i} = \begin{bmatrix}
h_{i1}(1) \\
\vdots \\
h_{i1}(P_{j})
\end{bmatrix}
\]

(27)

Like in Step 4, at \( p + 1 \) time point, the time origin changes from \( p \) to \( p + 1 \) time point, then the elements of vector \( \tilde{y}_{i,cor}(p + 1) \) should be moved, and the operation can be expressed by

\[
\tilde{y}_{i,NO}(p + 1) = S_{i,0}\tilde{y}_{i,cor}(p + 1),
\]

(28)

where

\[
S_{i,0} = \begin{bmatrix}
0 & 1 & 0 \\
\vdots & \ddots & \vdots \\
0 & 0 & 1
\end{bmatrix}
\]

(29)

Step 3. Model prediction based on fast model \( G_{td}(s) \).

On the t-time scale, the slow dynamic can be regarded as 0 which means that the slow dynamic changes very little. The model also can be fully expressed by information not so necessary as \( a_{ij} \). Let \( a_{ij} = [a_{i1}(1) \cdots a_{ij(N)}]^T, j = 1, 2 \). The fast predict model is

\[
\tilde{y}_{r,PM}(k) = \tilde{y}_{r,P0}(k) + A_{r}\Delta u, \quad j = 1, 2,
\]

(30)
where

\[
A_r = \begin{bmatrix}
  a_{r,1}(1) & 0 \\
  \vdots & \ddots \\
  a_{r,1}(M) & \cdots & a_{r,1}(1) \\
  \vdots & \ddots & \vdots \\
  a_{r,1}(P_f) & \cdots & a_{r,1}(P_f - M + 1)
\end{bmatrix}.
\] (31)

**Step 4.** Feedback Correction based on fast model \(G_i(s)\).

The input \(u(k)\) was applied to the plant \(G(s)\) at each \(k\) time point, and (22) gave the predictive output \(\hat{y}_{i,PM}(k+1 | k)\). Let error vector be

\[
e_e(k+1) = y_1(k+1) - \hat{y}_{i,PM}(k+1 | k),
\] (32)

where \(y_1(k+1)\) is the sample value. We used this error vector to modify the infection by some unsure factors, and we gave the error a weight vector to modify the prediction of output:

\[
\hat{y}_{i,cor}(k+1) = \hat{y}_{i,PM}(k) + H_r e_e(k+1),
\] (33)

where \(H_r = \begin{bmatrix} h_r(1) \\ \vdots \\ h_r(P_f) \end{bmatrix}\). At \(k+1\) time point, the time origin changes from \(k\) to \(k+1\) time point, then the elements of vector \(\hat{y}_{i,cor}(k+1)\) should be moved, and the operation can be expressed by

\[
\hat{y}_{i,PO}(k+1) = S_{r,0} \hat{y}_{i,cor}(k+1),
\] (34)

where

\[
S_{r,0} = \begin{bmatrix}
  0 & 1 & 0 \\
  \vdots & \ddots & \vdots \\
  \vdots & \ddots & \vdots \\
  0 & 1 & 0 \\
  0 & \cdots & 0
\end{bmatrix}.
\] (35)

**Step 5.** Rolling horizon optimization based on fast model \(G_i(s)\).

Let objective function be

\[
\min J(k) = \|\omega_1(k) - \hat{y}_{i,PM}(k)\|^2_{Q_1} + \|\omega_2(p) - \hat{y}_{i,PM}(p)\|^2_{Q_2} + \|\Delta u(k)\|^2_R
\] s.t. \(\Delta u_{\text{min}} \leq \Delta u \leq \Delta u_{\text{max}}\),

\[
u_{\text{min}} \leq u \leq u_{\text{max}},
\]

where \(\omega_i(k), i = 1, 2\), are the reference value:

\[
Q_1 = \text{diag}\left[q_1(1) \cdots q_1(P_f)\right],
\]

\[
Q_2 = \text{diag}\left[q_2(1) \cdots q_2(P_f)\right],
\]

\[
R = \text{block} - \text{diag}(R_1, R_2),
\]

\[
R_i = \text{diag}\left[r_i(1) \cdots r_i(M_f)\right], \quad i = 1, 2.
\] (37)

Without constraint, we can get the manipulated variables:

\[
\Delta u(k) = L \left[ (A_i^T Q_1 A_r + A_i^T Q_2 A_s + R)^{-1} \times \left[ A_i^T Q_1 (\omega_1(k) - \hat{y}_{i,PO}(k)) \right. \right.
\]

\[
\left. + A_i^T Q_2 (\omega_2(p) - \hat{y}_{i,PO}(p)) \right] \right\\
\]

where

\[
L = \begin{bmatrix}
  0 & \cdots & 0 & 0 \\
  \vdots & \ddots & \vdots & \vdots \\
  \vdots & \ddots & \vdots & \vdots \\
  0 & \cdots & 0 & 0 \\
  0 & \cdots & 0 & M
\end{bmatrix}.
\] (39)

Step 3 to step 5 form the inner circulate, and in every \(T_f\) time interval fast controller calculates a manipulate variable. Step 1 and Step 2 form the exterior circulate, and the slow controller provides the predictive value of the slow output to correct the fast controller every \(nT_f\) (\(n\) is a positive integer) time interval. So the framework of the control system was showed as shown in Figure 1.

\[\text{Figure 1: Framework of the control system.}\]

**4. Stability of the Two-Time Scale Decentralized MPC**

We introduced the algorithm step by step in the above sections. And we would like to discuss the stability of the controller in this section. First we put forward a sufficient condition of the controller with one inner circulate.

**Theorem 1.** A two-time scale decentralized MPC with the control parameters \(T_f, T_s\), and \(P_f\) and one inner circulate is stable under the sufficient condition of a standard MPC, which has a control interval \(T_f\) and a predictive horizon \(P_f\) (\(T_f/T_f\)).
Proof. Let the prediction horizon $P_f(T_f/T_f)$, be the manipulate horizon, let $M$ be the objective function of the standard MPC be

$$
\min J'(k) = ||(\omega(k) - \tilde{y}_{PM}(k))||_Q^2 + ||\Delta u(k)||_R^2
$$

s.t. $\Delta u_{\text{min}} \leq \Delta u \leq \Delta u_{\text{max}}$, \quad (40)

$$
\Delta u_{\text{min}} \leq u \leq \Delta u_{\text{max}},
$$

$$
J'(k) = ||(\omega(k) - \tilde{y}_{PM}(k))||_Q^2 + ||\Delta u(k)||_R^2
$$

$$
= ||\omega_1(k) - \tilde{y}_{1,PM}(k)||_{Q_1}^2
$$

$$
+ ||\omega_2(p) - \tilde{y}_{2,PM}(p)||_{Q_2}^2 + ||\Delta u(k)||_R^2
$$

$$
= \sum_{i=1}^{n} q_1'(\omega_1(k) - \tilde{y}_{1,PM}(i + k \mid k))^2
$$

$$
+ \sum_{i=1}^{n} q_2'(\omega_2(k) - \tilde{y}_{1,PM}(i + k \mid k))^2
$$

$$
+ \sum_{i=1}^{M} \Delta u(k + i \mid k)^2.
$$

Because the response speed of $y_1$ is very fast, the predictive value $\tilde{y}_{1,PM}(i + k \mid k)$ can be a fixed number when $i$ is larger than a certain number ($P_f$):

$$
P_f(T_f/T_f)
\sum_{i=1}^{n} q_1'(\omega_1(k) - \tilde{y}_{1,PM}(i + k \mid k))^2
$$

$$
= \sum_{i=1}^{n} q_1'(\omega_1(k) - \tilde{y}_{1,PM}(i + k \mid k))^2 + \text{Const.}
$$

(42)

The response speed of $y_2$ is very slow, so between a short time interval ($T_s$) the predictive values $\tilde{y}_{1,PM}(i + k \mid k)$, $C \leq i \leq C + (T_f/T_f)$ ($C$ is a positive number), can be linear correlation:

$$
P_f(T_f/T_f)
\sum_{i=C}^{C + (T_f/T_f)} q_2'(\omega_2(k) - \tilde{y}_{1,PM}(i + k \mid k))^2
$$

$$
= q_2'(\omega_2(k) - \tilde{y}_{1,PM}(C + k \mid k))^2,
$$

$$
P_f(T_f/T_f)
\sum_{i=1}^{n} q_2'(\omega_2(k) - \tilde{y}_{1,PM}(i + k \mid k))^2
$$

$$
= \sum_{j=1}^{n} q_2'(\omega_2(k) - \tilde{y}_{1,PM}(j + k \mid k)).
$$

Combining (38), (40), (42), and (44), we got

$$
J'(k) = J(k) + \text{const.}
$$

(45)

So $\min J(k)$ and $\min J'(k)$ had same answers, and the sufficient condition was proved.

When the inner circulate is larger than one, that is, $n > 1$, the algorithm can maintain its stability if the slow dynamic changes little in the time interval of the inner circulate. But it is hard to find an upper limit for $n$, because it is determined by the characteristic of the slow dynamic which can be quite different in different systems.

5. Case Study

A Model with Delay. We considered a two-in-two-out system. Two streams flow into a reactor, and $u_1$ and $u_2$ are the flow rates. The liquid level $y_1$ and the temperature $y_2$ are two controlled variables. $y_{10} = 50$ cm and $y_{20} = 295$ K are the initial stable states. The linear model of the system is

$$
\begin{bmatrix}
y_1 \\
y_2
\end{bmatrix} = G(s)
\begin{bmatrix}
u_1 \\
u_2
\end{bmatrix},
$$

(46)

$$
G(s) = \begin{bmatrix}
K_{11} & e^{-\tau_{21}s}
\\
K_{12} & e^{-\tau_{12}s}
\\
K_{21} & e^{-\tau_{21}s}
\\
K_{22} & e^{-\tau_{22}s}
\end{bmatrix},
$$

(47)

The response of output $y_1$ is much faster than that of $y_2$. If a standard DMC controller is applied on this system, the sample interval is determined by the fast response and should be very small, and the predictive horizon is determined by the slow response and should be very large. In such a small interval, it is difficult to calculate the optimal manipulate variables, and the control quality may be bad. If we compromise the sample time interval of different channels, we can get the following control effect to track step signals. Liquid level $y_1$ is set as 51 cm, and temperature is set as 296 K. Considering the uncertainty of the model, we chose the plant that each parameter above has 20% uncertainty to carry through the simulation. The plant is

$$
P(s) = \begin{bmatrix}
K_{11} + 0.2K_{11}e^{-(\tau_{11} + 0.2\tau_{11}s)}
\\
K_{12} + 0.2K_{12}e^{-(\tau_{12} + 0.2\tau_{12}s)}
\\
K_{21} + 0.2K_{21}e^{-(\tau_{21} + 0.2\tau_{21}s)}
\\
K_{22} + 0.2K_{22}e^{-(\tau_{22} + 0.2\tau_{22}s)}
\end{bmatrix},
$$

(48)

where $-1 < \epsilon < 1$ is a random number.

(1) $T = 0.2, P = 50$.

(2) $T = 10, P = 50$.

When the control interval is short, as seen from Figures 2 and 3, the maximum overshoot of fast channel is too big and the response speed of slow channel is too slow. The reason is that the predictive horizon is not long enough,
Table 2: Parameter values.

| Variable | Definition | Value |
|----------|------------|-------|
| $C_{A0}$ | Feed concentration of species A | 5.1 mol/L |
| $T_0$   | Feed temperature | 104.9 °C |
| $k_{10}$ | Collision factor for reaction 1: $k_{10}(T) = k_{10e}^{-E_1/T}$ | $1.287 \times 10^{15}$ h$^{-1}$ |
| $k_{20}$ | Collision factor for reaction 2: $k_{20}(T) = k_{20e}^{-E_2/T}$ | $1.287 \times 10^{15}$ h$^{-1}$ |
| $k_{30}$ | Collision factor for reaction 3: $k_{30}(T) = k_{30e}^{-E_3/T}$ | $9.043 \times 10^{4} (mol \ A)^{-1} h^{-1}$ |
| $E_1$   | Normalized activation energy for reaction 1 | $-9758.3$ K |
| $E_2$   | Normalized activation energy for reaction 1 | $-9758.3$ K |
| $E_3$   | Normalized activation energy for reaction 1 | $-8560$ K |
| $\Delta H_{RAB}$ | Enthalpies of reaction 1 | 4.2 kJ/mol A |
| $\Delta H_{RAB}$ | Enthalpies of reaction 2 | $-11$ kJ/mol B |
| $\Delta H_{RAB}$ | Enthalpies of reaction 3 | $-41.85$ kJ/mol A |
| $k_w$   | Heat transfer coefficient for cooling jacket | $4.032$ kJ/(h m$^2$ K) |
| $A_R$   | Surface of cooling jacket | 0.215 m$^2$ |
| $V_R$   | Reactor volume | 0.01 m$^3$ |
| $m_K$   | Coolant mass | 5.0 kg |
| $C_{PK}$ | Heat capacity of coolant | 2.00 kJ/(kg·K) |
| $C_p$   | Heat capacity | 3.01 kJ/(kg·K) |
| $\rho$  | Density | 0.9342 kg/L |

Table 3: Stable states.

| $C_A$   | $C_B$   | $T$      | $T_k$   | $u_1$ | $u_2$ |
|---------|---------|----------|---------|-------|-------|
| 2.4308 mol/L | 1.0802 mol/L | 115.4559 C | 114.9944 C | 20 min$^{-1}$ | $-400$ kJ/h |

Figure 2: Response of $y_1$.  
Figure 3: Response of $y_2$.  

The prediction model is not fully used, and the feedback correction plays an important role in this set point tracking process. When the control interval is long, as seen from Figures 4 and 5, the slow channel shows good control quality, but the respond speed of fast channel is still a little slow. A long control interval means a low control frequency and the fast channel achieves its set point by several control steps, so the fast channel response is a little slow. And the fast channel can achieve a temporary stable state, so the response curve may have a stair shape. The compromise methods are not so perfect. Therefore, we designed a decentralized DMC controller, and let $T_f = 0.2$, $T_s = 20$, $n = 5$, $P_s = 50$, and $P_f = 50$. We used this decentralized controller to track step signals. As shown in Figures 6 and 7, both fast channel and slow channel showed very good control quality. This method combined fast dynamic information and stable state information. The model information was fully used. From the simulation, we found that the decentralized method showed better control quality than the two compromised method.
Figure 4: Response of $y_1$.

Figure 5: Response of $y_2$.

Figure 6: Response of $y_1$.

Figure 7: Response of $y_2$.

Figure 8: Response of $C_A$.

Figure 9: Response of $T_k$. 
A Continuously Stirred Tank Reactor (CSTR) Model. Chen studied the nonlinearity of a CSTR and modeled the CSTR by the following nonlinear equations [20]:

\[
\begin{align*}
\frac{dC_A}{dt} &= \frac{\dot{V}}{V_R}(C_{A0} - C_A) - k_1(T)C_A - k_3(T)C_A^2, \\
\frac{dC_B}{dt} &= -\frac{\dot{V}}{V_R}C_B + k_1(T)C_A - k_2(T)C_B, \\
\frac{dT}{dt} &= \frac{\dot{V}}{V_R}(T_0 - T) \\
&\quad - \frac{1}{\rho C_p}(k_1(T)C_A\Delta H_{RAB} + k_2(T)C_B\Delta H_{RBC}) \\
&\quad + k_3(T)C_A^2\Delta H_{RAD} + \frac{k_wA_R}{\rho C_p V_R}(T_K - T), \\
\frac{dT_K}{dt} &= \frac{1}{m_KC_{pK}}(Q_K + k_wA_R(T - T_K)).
\end{align*}
\]  

(49)

We chose \( u_1 = \dot{V}/V_R \) and \( u_2 = Q_K \) as the manipulate variables, and \( C_A \) and \( T_K \) as the controlled variables. Parameter values were given in Table 2.

We chose a steady state and identified the input-output model to design a model predictive controller.

The input-output model is

\[
\begin{bmatrix}
C_A \\
T_K
\end{bmatrix} = 
\begin{bmatrix}
\begin{bmatrix}
2.868 \\
-3.633 \\
0.04559
\end{bmatrix} & \begin{bmatrix}
0 \\
0 \quad 0.04559 \\
\end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2
\end{bmatrix},
\]

(50)

The respond speed of \( T_K \) is a little faster than that of \( C_A \). And the sample frequency of the two kinds of sensors cannot be same due to the limits of the sensors. The temperature sensor can only be sampled in high frequency, while the concentration sensor can only be sampled in low frequency. In order to maintain high control frequency, soft-sensing methods are often used in standard MPC. Due to computational errors and other errors, the soft-sensing method is
not a perfect way. The decentralized method presented in this paper can also deal with different sample frequencies in different channels. We compared 0.1 step tracing effect of the standard MPC and that of the decentralized method. Let the sample interval of the temperature sensor be 0.1 minute, and let the sample interval of the concentration sensor be 1 minute.

1. Perfect soft-sensing, \( T = 0.1 \) minute, \( P = 50 \).
2. Without soft-sensing \( T = 1 \) minute, \( P = 50 \).
3. Decentralized MPC \( T_f = 0.1 \) minute, \( T_s = 2 \) minutes, \( n = 10 \).

For the systems with little dynamic differences in different channels, if the controlled variables can provide sufficient reliable information in high frequency, the standard MPC can be applied and good control quality can be achieved (Figures 8 and 9). If the controlled variables can only provide reliable information in low frequency, the tracking speed turns slow (Figures 10 and 11). Although the dynamic characteristics of the two channels have a few differences, the decentralized MPC can also achieve good control quality (Figures 12 and 13) because the decentralized method takes full advantage of the reliable information of different channels.

6. Conclusion

In this article, we focused on a kind of special system and designed a decentralized model predictive controller for it. This kind of system has different dynamics in different channels and exhibits two-time scale. A centralized MPC controller cannot satisfy the fast and the slow channel simultaneously. We used singular perturbation method to get the fast and the slow model from the original system. In actual processes, input-output models that can be obtained easily by identification were usually used to describe the real system. We demonstrated the singular perturbation method applying in transfer function matrix. Then we presented a decentralized model predictive controller based on the fast and the slow model and provided a sufficient condition for the algorithm stability when \( n = 1 \). Finally, the decentralized model predictive control algorithm was applied in two examples by simulation, and the validity of the control algorithm was tested. The simulation results proved that the two-time scale MPC is superior to the traditional MPC when the system had two-time scale characteristic.

The algorithm is based on the idea of fully using the information of the system. For the systems with two-time scale characteristics, the fast and slow channels are controlled, respectively, in the decentralized algorithm. This algorithm makes best use of the transition information of the fast channels and the slow channels and reduces the computation burden, which provides short control interval and increases the response speed. For those systems without two-time scale characteristics, this algorithm also works well. MPC has intensively been applied in the industrial process. The two-time scale MPC algorithm which is presented in this paper extends the applying scopes of MPC.

Acknowledgments

The authors gratefully acknowledge the financial support of 863 Program of China (no. 2007AA041402), National Key Scientific and Technical Project of China (no. 2007BAF22B05) and National Science Foundation of China (no. 60804023).

References

[1] P. V. Kokotovic, R. E. O’Malley Jr., and P. Sannuti, “Singular perturbations and order reduction in control theory—an overview,” Automatica, vol. 12, no. 2, pp. 123–132, 1976.
[2] R. G. Phillips, “A two-stage design of linear feedback controls,” IEEE Transactions on Automatic Control, vol. 25, no. 6, pp. 1220–1223, 1980.
[3] A. Saberi and H. K. Khalil, “Two time-scale feedback design of nonlinear singularly perturbed system,” in Proceedings of the American Control Conference (ACC ’83), pp. 441–442, San Francisco, Calif, USA, June 1983.
[4] P. V. Kokotovic, “Subsystems, time scales and multimodeling,” Automatica, vol. 17, no. 6, pp. 789–795, 1981.
[5] H. K. Khalil and A. Saberi, “Decentralized stabilization of nonlinear interconnected systems using high gain feedback,” IEEE Transactions on Automatic Control, vol. 27, no. 1, pp. 265–268, 1982.
[6] C. K. Yi and W. L. Luyben, “Design and control of coupled reactor/column systems—part 1: a binary coupled reactor/rectifier system,” Computers and Chemical Engineering, vol. 21, no. 1, pp. 25–46, 1996.
[7] C. K. Yi and W. L. Luyben, “Design and control of coupled reactor/column systems—part 2: more complex coupled reactor/column systems,” Computers and Chemical Engineering, vol. 21, no. 1, pp. 47–67, 1996.
[8] C. K. Yi and W. L. Luyben, “Design and control of coupled reactor/column systems—part 3: a reactor/stripper with two columns and recycle,” Computers and Chemical Engineering, vol. 21, no. 1, pp. 69–86, 1996.
[9] M.-N. Contou-Carrere and P. Daoutidis, “Dynamic precompensation and output feedback control of integrated process networks,” in Proceedings of the American Control Conference (ACC ’94), vol. 4, pp. 2909–2914, Boston, Mass, USA, June–July 2004.
[10] A. Kumar and P. Daoutidis, “Nonlinear dynamics and control of process systems with recycle,” Journal of Process Control, vol. 12, no. 4, pp. 475–484, 2002.
[11] N. Vora and P. Daoutidis, “Nonlinear model reduction of chemical reaction systems,” in Proceedings of the American Control Conference (ACC ’99), vol. 3, pp. 1583–1587, San Diego, Calif, USA, June 1999.
[12] A. Kumar, P. D. Christofides, and P. Daoutidis, “Singular perturbation modeling of nonlinear processes with nonexplicit time-scale multiplicity,” Chemical Engineering Science, vol. 53, no. 8, pp. 1491–1504, 1998.
[13] J. M. Maciejowski, Predictive Control with Constraints, Prentice Hall, London, UK, 2002.
[14] K. L. Buescher and C. C. Baum, “A two-time-scale approach to nonlinear model predictive control,” in Proceedings of the American Control Conference (ACC ’95), vol. 3, pp. 2250–2256, Seattle, Wash, USA, June 1995.
[15] H. K. Khalil, “Output feedback control of linear two-time-scale system,” IEEE Transactions on Automatic Control, vol. 32, no. 9, pp. 784–792, 1987.
[16] S. F. Wang and J. H. Du, *Illustrative Examples and Problems of Automatic Control Theory*, Tsinghua University Press, Beijing, China, 2002.

[17] Y. G. Xi, *Introduction on Dynamic Large-Scale Systems*, National Defense Industry Press, Beijing, China, 1988.

[18] R. S. Madriz and S. S. Sastry, “Input-output description of linear systems with multiple time-scales,” *International Journal of Control*, vol. 40, no. 4, pp. 699–721, 1984.

[19] Y. G. Xi, *Predictive Control*, National Defense Industry Press, Beijing, China, 1993.

[20] H. Chen, A. Kremling, and F. Allgower, “Nonlinear predictive control of a benchmark CSTR,” in *Proceedings of the 3rd European Control Conference (ECC ’95)*, pp. 3247–3252, Rome, Italy, 1995.