Adaptive General Predictive Control Using Optimally Scheduled Multiple Models for Parallel-Coursing Utility Units With a Header

Lei Pan[1]
e-mail: panlei@seu.edu.cn

Jiong Shen

School of Energy and Environment,
Southeast University,
Nanjing 210096, China

Peter B. Luh
Department of Electrical and
Computer Engineering,
University of Connecticut,
Storrs, CT 06269

An adaptive general predictive control using optimally scheduled multiple models (OSMM-GPC) is presented for improving the load-following capability and economic profits of the system of parallel-coursing utility units with a header (PUUH). OSMM-GPC is a comprehensive control algorithm built on the distributed multiple-model control architecture. It is improved from general predictive control by two novel algorithms. One is the mixed fuzzy recursive least-squares (MFRLS) estimation and the other is the model optimally scheduling algorithm. The MFRLS mixes the local and global online estimations by weighting a dynamic multi-objective cost function on the membership feature of each sampling point. It provides better parameter estimation on the Takagi–Sugeno (TS) fuzzy model of a time-varying system than the local and global recursive least squares, thus, it is proper for building adaptive models for the OSMM-GPC. Based on high-precision adaptive models estimated by the MFRLS, the model optimally scheduling algorithm computes the regulating efficiencies of all control groups and then chooses the optimal one in charge of the multiple-variable general predictive control. Through the model scheduling at each operation point, considerable fuel consumption can be saved; meanwhile, a better control performance is achieved. Besides PUUH, the OSMM-GPC can also work for other distributed multiple-model control applications.
[DOI: 10.1115/1.4006085]

Keywords: adaptive model, general predictive control, fuzzy multiple model, thermal power plant, pressure control

1 Introduction

The system of PUUH is an important form of combined generation of electricity and heat in refineries, chemical plants, and paper mills. A PUUH system is composed of several source devices like fossil-fuel boilers and several sink devices like steam turbines connected by a steam header as shown in Fig. 1. The main purpose of PUUH control is to keep balance between the load demand and the supply of the overall system. Load demands of PUUH come from two ways: One is the static load demand from the upper management level to the control level. It can be allocated by a variety of optimization approaches, which we have studied before [1–3]; another is the dynamic load demand (DLD) of steam flow from heat users. It is shown as a real-time decrease or increase of the header pressures proportional to the backpressures of the steam turbines on the header; therefore, an optimal allocation of DLD is equivalent to an optimal control of header pressures with the objective to minimize fuel consumption.

For the requirement of stable steam quality, the controller should keep the measured header pressures at their set points precisely in steady states and fluctuate shortly and smoothly in transient processes. However, header pressures are very difficult to regulate, because they are influenced by multiple sources and multiple sinks, resulting in a complicated coupled system with long delays. When a disturbance occurs, it usually takes a PUUH system long time to arrive at a new stable state, showing poor stability and large fuel consumption. These control problems become more difficult when DLD varies in a wide range with nonlinear variations of the coal consumption rates of the boilers, and when the parameters of PUUH vary slowly in the long-term operation. Thus far, the header-pressure regulating is a bottleneck in improving the load-following capability and the generation efficiency of a PUUH system.

From the literature and large numbers of field operation reports, we know that the existing header-pressure controllers with the conventional algorithm do not work well in practice. There have been some studies on improving header-pressure control in its stability and reliability [4–7], but none of them consider the optimal fuel consumptions during their pressure regulation. Therefore, an improved header-pressure control with the DLD optimal allocating needs to be developed under the objective of the minimal fuel consumption currently.

General predictive control is a receding-horizon linear quadratic control law depending on the prediction of the plant's output over several steps. It has been very successful in solving the control problems with long delay, with coupled inputs, and with variable parameters [8–10]. Although general predictive control algorithm is a linear control law, it can be extended to nonlinear control by incorporating nonlinear prediction models.

We have proposed a theoretical approach for modeling PUUH [11]. Because it has some recurrent computing, it is suitable for building a simulator but not for serving as a prediction control model. The Takagi–Sugeno fuzzy model is an effective multiple-model approximate method for modeling complex nonlinear systems [12]. It has shown its good effects on improving the

[1]Corresponding author.
Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement and Control. Manuscript received May 15, 2011; final manuscript received December 29, 2011; published online April 26, 2012. Assoc. Editor: Won-jong Kim.

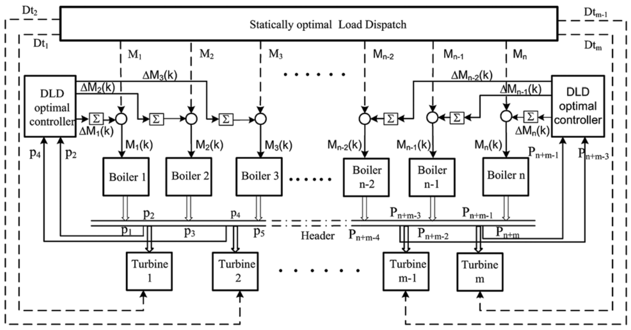

Fig. 1 PUUH with control system

performance of many nonlinear multiple-model control systems and thus has been widely applied [13–17]. In order to improve the header-pressure control performance as well as to realize the DLD optimal allocating of PUUH in wide operation ranges, we will develop an adaptive OSMM-GPC based on the TS fuzzy models in this paper.

In the novel OSMM-GPC, the general predictive control for header pressure is improved by two new algorithms. One is the MFRLS estimation for achieving a high precision of online-estimated models of nonlinear and time-varying systems. It contributes to getting correct scheduling and predictions in the OSMM-GPC. Another model, optimally scheduling algorithm improves the multiple-model control into a distributed multiple-model control by a two-step model scheduling. It first sorts out the boiler combination of the highest regulating efficiency from the TS model set estimated by MFRLS, and then it schedules the active prediction model by fuzzily weighting the TS models of the chosen combination. Based on the active prediction model, the general predictive control algorithm works out the optimal control instructions and then sends them to the chosen boilers at each sampling instant. OSMM-GPC can realize the DLD optimal allocating of PUUH by considering both the economy index and regulatory index in its two-step model scheduling. It provides a better multi-objective optimal control solution to distributed nonlinear systems than other multiple-model control strategies by its more flexible optimally scheduling algorithm.

This paper is organized in the following way: Sec. 2 describes the distributed increment-control architecture in which the OSMM-GPC works. Section 3 describes the modeling approach. First, there is a brief introduction of the TS model and then the derivation of the MFRLS algorithm. Section 4 proposes the model optimally scheduling algorithm first, then it presents the wholly OSMM-GPC. The simulation experiments in Sec. 5 verified both MFRLS on the multiple-variable plant and the OSMM-GPC algorithms.

2 The Distributed Increment-Control Architecture

Figure 1 shows the distributed increment-control architecture of a PUUH system with n boilers and m steam turbines ($n > m$). $M_1 - M_n$ denotes the allocated static boiler load (firing rates) and $Dt_1 - Dt_m$ the allocated static steam turbine load (steam flow). They are the basic supply and demand loads, respectively. We only show their relation to DLD in the architecture and will not discuss them more. $\Delta M_1 - \Delta M_n$ denotes the real-time varying DLD optimally allocated by the DLD optimal controllers or the optimal header-pressure controllers to be discussed under the objective of the minimal fuel consumptions.

The DLD optimal controller adopts multiple-variable control algorithm. Considering the computing complexity, a distributed control system with several low-dimension DLD controllers is more proper than one controller for a whole system of PUUH. The number of controllers depends on the size of a PUUH. Intuitively, the DLD optimal controllers of a distributed system can be set according to the design principle of a PUUH system. It is to compose the overall system of several flow demand-and-supply balanceable groups according to the nominated capacities and locations of devices. A 3-boiler and 2-turbine group is a typically configured balanceable group. The flow demand-and-supply balance can be kept within one group. The flow between two neighbored groups is small thus can be treated as a disturbance for each group in control. Therefore, we can design a distributed control system for PUUH by setting one DLD optimal controller in each balanceable group as shown in Fig. 1.

Each DLD optimal controller has multiple inputs and multiple outputs. In Fig. 1, P_i, $i = 1, \ldots, n+m-1$, denotes the pressure measurement on the ith header segment, which is a portion of the header connected with the ith device. Because they can sensitively show the unbalance between the load demand and supply in their group, the variations of P_i on the header segments with steam turbines can be used as the controlled variables of the DLD optimal control. Obviously, the control variables should be the incremental firing rate of boilers $\Delta M_i(k)$, $i = 1, \ldots, n$. Take a PUUH with $n = 3$ and $m = 2$ as an example. The inputs of the DLD optimal controller are P_2 and P_4, which are the controlled variables. Its outputs are the increment firing rates $\Delta M_1(k)$–$\Delta M_3(k)$ sent to the boilers 1–3 as the control variables.

Because the control variables are incremental, this distributed increment-control architecture supports a freely and smoothly scheduling on control variables among different boilers. Therefore, we will study OSMM-GPC algorithm on this architecture which may not only be suitable for PUUH but also for other bus-topology systems which need distributed multiple-model control.

3 Takagi-Sugeno Fuzzy Modeling With MFRLS Estimation

The performance of OSMM-GPC largely depends on how precisely the online-estimated TS model tracks the nonlinear and slow time-varying system; thus, it is valuable to improve the online estimation precision of TS models. Many fuzzy clustering methods have been well established for the estimation of the antecedent parameters of TS models, e.g., fuzzy C-Mean clustering, Gustafson–Kessel clustering, Gath–Geva clustering, and the online clustering methods [18–20], thus we will not discuss them here.

The most widely used estimation method on consequent parameters of TS models is the fuzzy least-squares estimation [21]. Although the fuzzy batch least squares for offline estimation have been well studied [22], the more valuable online estimation method is the fuzzy recursive least squares (FRLS) for tracking time-varying parameters of real systems. The existing FRLS algorithms include the global optimal FRLS and the local optimal FRLS [23]. The former pursues the global fitting but ignores the interpretability of each local model, whereas the latter has good interpretability on local linear models but may not achieve the optimal global approximation. Therefore, both the two FRLS algorithms do not behave ideally in tracking the time-varying consequent parameters of TS models as seen in the experiments of Sec. 5. The reason is that they use a fixed tracking objective without considering the varying feature (membership degree) of each newly coming sampling point in their online learning. If one new sampling point has a high membership degree in one fuzzy partition and low membership degrees in other fuzzy partitions, the point should mainly contribute to the local model learning because it belongs to a predominant subspace; else, if the point has the almost even membership degree in every partition, it is suitable for the global learning. Weighting the two FRLSs and computing in parameter estimation to improve the precision may be intuitively thought of, but actually it does not work. Considering the features of sampling points in online learning, we proposed a novel FRLS on both global and local optimal estimation objectives. This method uses the feature of each new sampling point to adjust the objective for the optimal parameter estimations at each updating instant. The recursion formula is obtained by solving a dynamic multiple-objective optimization problem at each sampling instant. In the novel FRLS, the global and local FRLSs are mixed by their common covariance matrix, so we called it the MFRLS algorithm, the formulas were presented in Ref. [17] without derivation. In this paper, we will give a complete presentation on how to solve the mixed covariance matrix on the multiple optimal objectives and derive the recursion formula of MFRLS which has not been published before. First, a brief introduction of TS model will be given before the derivation of the MFRLS algorithm.

3.1 Takagi-Sugeno Fuzzy Model. TS model is capable of describing highly nonlinear system using a group of rule-based models with fuzzy antecedents and functional consequents that follow the fuzzy inference proposed by Takagi and Sugeno [21]. A typical rule-based TS model is shown as

$$\begin{aligned} &R_i : \text{If } (\psi_1(k) \text{ is } \Omega_{i,1}) \text{ and } \ldots \text{ and } (\psi_s(k) \text{ is } \Omega_{i,s}) \\ &\text{Then } y_i(k) = \theta_{i,1}\psi_1(k) + \ldots + \theta_{i,s}\psi_s(k) + \theta_{i,s+1};\ i = 1, 2, \ldots, K_l \end{aligned} \tag{1}$$

where R_i denotes the ith fuzzy rule; k is the kth sampling time instant; K_l is the number of fuzzy rules; $\Omega_{i,j}$ denotes the antecedent fuzzy sets, $j = 1,\ldots,S$; ψ is the input vector, $\psi(k) = [\psi_1(k),\ldots,\psi_s(k)]^{\mathrm{T}} \in R^{\mathrm{S}}$, here $\psi_j(k)$, denotes the jth scalar of vector ψ; $\theta_{i,j}$, $j = 1,\ldots,S+1$, is the coefficients of the consequent polynomials to be identified; y_i is the output of the ith linear subsystem. The output of the TS model is computed by

$$y(k) = \sum_{i=1}^{k_l} \mu_i(k)(\theta_{i,1}\psi_1(k) + \cdots + \theta_{i,s}\psi_s(k) + \theta_{i,s+1}) \tag{2}$$

where the normalized firing level of rule R_i is defined as

$$\mu_i(k) = \frac{\omega_i(k)}{\sum_{i=1}^{k_l} \omega_i(k)} \tag{3}$$

ω_i is the firing degree of rule R_i at instant k, which is defined as conjunction of respective fuzzy set for this rule

$$\omega_i(k) = \Omega_{i,1}(\psi_1(k)) \times \Omega_{i,2}(\psi_2(k)) \times \cdots \times \Omega_{i,s}(\psi_s(k)) \tag{4}$$

where $\Omega_{i,j}(\psi_j(k))$, $j = 1, \ldots, S$, is the membership degree of $\psi_j(k)$ in the fuzzy set $\Omega_{i,j}$.

3.2 MFRLS Estimation on Consequent Parameters. We take both the global estimation error and the local estimation error into account to develop MFRLS for more precise estimation of consequent parameters. The key point of MFRLS is to deduce the mixed covariance matrix of its recursive formula associated with the varying member degree of the latest sampling point at each sampling instant.

First, we need to construct the same dimensional matrices and vectors for dynamically building the multi-objective cost function composed of the global and local estimation errors. Assume, there are N groups of input–output sampling point data

$$(\psi_e(k), \bar{y}(k)), \quad k = 1, \ldots, N \tag{5}$$

where

$$\psi_e(k) = [\psi_1(k), \ldots, \psi_s(k), 1]^T \in R^{S+1} \tag{6}$$

and $\bar{y}(k)$ is the sampled output at instant k. Transform (2) into vector form on the N groups of data

$$Y = \Psi^{\mathrm{T}}\theta \tag{7}$$

where

$$\theta = [\theta_1{}^{\mathrm{T}}, \ldots, \theta_{Kl}{}^{\mathrm{T}}]^{\mathrm{T}} \tag{8}$$

$$\theta_i = [\theta_{i,1}, \ldots, \theta_{i,s+1}]^{\mathrm{T}}, \quad i = 1, \ldots, K_l \tag{9}$$

$$\Psi = \begin{pmatrix} \mu_1(1)\psi_e^T(1) & \mu_2(1)\psi_e^T(1) & \cdots & \mu_{K_l}(1)\psi_e^T(1) \\ \mu_1(2)\psi_e^T(2) & \mu_2(2)\psi_e^T(2) & \cdots & \mu_{K_l}(2)\psi_e^T(2) \\ \cdots & \cdots & \cdots & \cdots \\ \mu_1(N)\psi_e^T(N) & \mu_2(N)\psi_e^T(N) & \cdots & \mu_{K_l}(N)\psi_e^T(N) \end{pmatrix}_{N\times[K_l\times(S+1)]} \tag{10}$$

$$Y = [\bar{y}(1), \ldots, \bar{y}(N)]^T \tag{11}$$

For the purpose of making a multi-objective function of the global and local estimation errors with respect to θ, we construct the following matrix and vectors:

$$\tilde{Y} = \begin{pmatrix} Y^T & Y^T & \cdots & Y^T \end{pmatrix} T \in R^{(N\times K_l)\times 1} \tag{12}$$

$$\Lambda = \mathrm{diag}(\Lambda_1, \cdots, \Lambda_{K_l}) \in R^{(N\times K_l)\times(N\times K_l)} \tag{13}$$

$$X = \begin{pmatrix} \psi_e^T(1) & & \\ \vdots & 0 & 0 \\ \psi_e^T(N) & & \\ 0 & \ddots & 0 \\ & & \psi_e^T(1) \\ 0 & 0 & \\ & & \psi_e^T(N) \end{pmatrix} = \begin{pmatrix} x_{1,1}^T \\ \vdots \\ x_{1,N}^T \\ \vdots \\ x_{K_l,1}^T \\ \vdots \\ x_{K_l,N}^T \end{pmatrix}_{(N\times K_l)\times[K_l\times(S+1)]} \tag{14}$$

define $X_i = [\psi_e\ (1), \psi_e\ (2) ,\ldots, \psi_e\ (N)]^{\mathrm{T}} \in R^{N\times(S+1)}$, $i = 1,\ldots, K_l$, is the ith diagonal block matrix of X; the vector $x^{\mathrm{T}}_{i,j} \in R^{1\times[(S+1)\times Kl]}$, $i = 1,\ldots,K_l$, $j = 1,\ldots,N$, denotes the row vector of X including the jth row vector of X_i. $\hat{\theta}$ is the estimation of θ.

Second, we make a multi-objective cost function based on Eqs. (5)–(14), by the weighted summation of the squares of the global and local estimation errors. The mixed objective function is given by

$$J_m = \bar{\mu}_m(k)(Y - \Psi\hat{\theta})^T(Y - \Psi\hat{\theta}) + \mu_m(k)\,(\tilde{Y} - X\hat{\theta})^T\Lambda(\tilde{Y} - X\hat{\theta}) \tag{15}$$

where the first item is the weighted global estimation error and the second item is the weighted local estimation error; where $\bar{\mu}_m(k) + \mu_m(k) = 1$, and $\mu_m(k)$ is the highest normalized firing level among all the submodels of the latest sampling point at instant k.

Third, we minimize the objective function (15) with respect to $\hat{\theta}$ and then deduce the recursive formula of MFRLS. The mixed optimal (least-squares) estimation of θ is given by

$$\theta = C_m(k)[\bar{\mu}_m(k)\Psi^T Y + \mu_m(k)X^T\Lambda\tilde{Y}] \tag{16}$$

$$C_m(k) = [\bar{\mu}_m(k)\Psi^T\Psi + \mu_m(k)X^T\Lambda X]^{-1} \tag{17}$$

where $C_m(k)$ is the $[K_l \times (S+1)] \times [K_l \times (S+1)]$ covariance matrix. In Eq. (17), let $C_G = (\psi^{\mathrm{T}}\psi)^{-1}$, which is the global estimation covariance matrix, and the recursive formula of C_G is given by

$$C_G(k) = C_G(k-1) - \frac{C_G(k-1)\Psi_k\Psi_k^T C_G(k-1)}{1 + \Psi_k^T C_G(k-1)\Psi_k} \tag{18}$$

where $\Psi_k{}^{\mathrm{T}} = [\mu_1(k)\psi_e{}^{\mathrm{T}}(k),\ldots, \mu_{Kl}(k)\psi_e{}^{\mathrm{T}}(k)] \in R^{1\times[Kl\times(S+1)]}$. Let $C_L = (X^{\mathrm{T}}\Lambda X)^{-1}$, which is the local estimation covariance matrix. We derive the recursive formula of C_L as follows:

$$C_L(k) = (\mathrm{X}^T \Lambda \mathrm{X})^{-1} = \left(\sum_{j=1}^{k} \sum_{i=1}^{k_l} \mu_i(j) x_{i,j} x_{i,j}^T \right)^{-1}$$
$$= \left(\sum_{j=1}^{k-1} \sum_{i=1}^{k_l} \mu_i(j) x_{i,j} x_{i,j}^T + \sum_{i=1}^{k_l} \mu_i(k) x_{i,k} x_{i,k}^T \right)^{-1} \tag{19}$$
$$= \left[C_L{}^{-1}(k-1) + \sum_{i=1}^{k_l} \mu_i(k) x_{i,k} x_{i,k}^T \right]^{-1}$$

Applying the matrix inversion lemma on Eq. (19), we have the recursive formula of C_L

$$C_L(k) = C_L(k-1) - \frac{C_L(k-1) \sum_{i=1}^{K_l} \mu_i(k) x_{i,k} x_{i,k}^T C_L(k-1)}{I + C_L(k-1) \sum_{i=1}^{K_l} \mu_i(k) x_{i,k} x_{i,k}^T} \tag{20}$$

where $x_{i,k}$ is defined in matrix X in Eq. (14). From Eq. (17), we have

$$C_m{}^{-1}(k) = \bar{\mu}_m(k) C_G^{-1}(k) + \mu_m(k) C_L^{-1}(k)$$
$$= \bar{\mu}_m(k) \cdot \left[C_G{}^{-1}(k-1) + \Psi_k \Psi_k^T \right]$$
$$+ \mu_m(k) \cdot \left[C_L{}^{-1}(k-1) + \sum_{i=1}^{k_l} \mu_i(k) x_{i,k} x_{i,k}^T \right]$$
$$= C_m{}^{-1}(k-1) + \bar{\mu}_m(k) \cdot \Psi_k \Psi_k^T + \mu_m(k) \cdot \sum_{i=1}^{k_l} \mu_i(k) x_{i,k} x_{i,k}^T \tag{21}$$

and

$$C_m{}^{-1}(k-1) = C_m{}^{-1}(k) - \bar{\mu}_m(k) \cdot \Psi_k \Psi_k^T + \mu_m(k) \cdot \sum_{i=1}^{k_l} \mu_i(k) x_{i,k} x_{i,k}^T \tag{22}$$

Transform (16) into the summation form and simplify it by

$$\theta(k) = C_m(k) \cdot [\bar{\mu}_m(k) \cdot \sum_{j=1}^{k} \Psi_j \bar{y}(j) + \mu_m(k) \cdot \sum_{j=1}^{k} \sum_{i=1}^{k_l} \mu_i(j) x_{i,j} \bar{y}(j)]$$
$$= C_m(k) \cdot \{\bar{\mu}_m(k) \cdot \sum_{j=1}^{k-1} \Psi_j \bar{y}(j) + \mu_m(k) \cdot \sum_{j=1}^{k-1} \sum_{i=1}^{k_l} \mu_i(j) x_{i,j} \bar{y}(j)$$
$$+ \bar{\mu}_m(k) \cdot \Psi_k \bar{y}(k) + \mu_m(k) \cdot \sum_{i=1}^{k_l} \mu_i(k) x_{i,k} \bar{y}(k)\}$$
$$= C_m(k) \cdot \{C_m{}^{-1}(k-1) \cdot \theta(k-1) + \bar{\mu}_m(k) \cdot \Psi_k \bar{y}(k)$$
$$+ \mu_m(k) \cdot \sum_{i=1}^{k_l} \mu_i(k) x_{i,k} \bar{y}(k)\} \tag{23}$$

Substituting Eq. (22) in Eq. (23), we have

$$\theta(k) = \theta(k-1) + C_m(k)[\bar{\mu}_m(k) \cdot \Psi_k \bar{y}(k)$$
$$+ \mu_m(k) \cdot \sum_{i=1}^{K_l} \mu_i(k) x_{i,k} \bar{y}(k) - \bar{\mu}_m(k) \cdot \Psi_k \Psi_k^T \theta(k-1)$$
$$- \mu_m(k) \cdot \sum_{i=1}^{K_l} \mu_i(k) x_{i,k} x_{i,k}^T \theta(k-1)]$$
$$= \theta(k-1) + C_m(k)\{\bar{\mu}_m(k) \cdot \Psi_k[\bar{y}(k) - \hat{y}(k)]$$
$$+ \mu_m(k) \cdot \sum_{i=1}^{K_l} \mu_i(k) x_{i,k} [\bar{y}(k) - \hat{y}_i(k)]\} \tag{24}$$

which is shortly written as

$$\theta(k) = \theta(k-1) + C_m(k)\{\bar{\mu}_m(k) \cdot \Psi_k[\bar{y}(k) - \hat{y}(k)]$$
$$+ \mu_m(k) \cdot \sum_{i=1}^{K_l} \mu_i(k) x_{i,k} [\bar{y}(k) - \hat{y}_i(k)]\} \tag{25}$$

where $\hat{y}(k) = \Psi_k^T \theta(k-1)$ is the global estimation of $\bar{y}(k)$; $\hat{y}_i(k) = x_{i,k}^T \theta(k-1)$ is the ith local estimation of $\bar{y}(k)$.

Finally, we have the MFRLS algorithm which is described by the recursive formulas (18), (20), (21), and (25). Although we derive MFRLS from the batch global and local least-squares estimations, the resultant formulas can work only with the current sampling data and their outputs of the previous sampling instant. In the MFRLS, the firing level of a new sampling point is calculated and then the maximal one among all the subspaces is chosen to construct a multi-objective optimization index for updating the consequent parameters at each sampling step; thus, MFRLS is a more effective and flexible estimation method than the other two FRLS.

Based on nonlinear and time-varying prediction models achieved by the TS fuzzy modeling with MFRLS, we are able to study the optimally scheduling algorithm on multiple prediction models for the general predictive control.

4 The OSMM-GPC Algorithm

The purpose of an optimal control of header pressures or the DLD optimal controller is to track DLD efficiently, while achieving a better control performance including smaller overshoot, setting time, and steady-state errors on the header pressures. We will propose the OSMM-GPC to improve the control performance of header pressures, the DLD-following capability and the economic profit of PUUH in this section.

Suppose there are n boilers and m steam turbines working in one balanceable group of PUUH. In most cases, there is $n > m$, therefore, a model optimally scheduling algorithm is needed for selecting the m-most efficient boilers to be the regulating devices of the $m \times m$ general predictive control on the header pressures. Because of the nonlinearity and time-varying characteristics of PUUH, the selection should be online executed at each operation point. As mentioned previously, a smooth switching of control devices can be guaranteed on the increment-control architecture in Fig. 1.

The DLD optimal controller shown in Fig. 2 uses the OSMM-GPC algorithm to implement a distributed control on the pressures along the header. The OSMM-GPC algorithm has three major functions including the modeling and online estimating by Takagi–Sugeno fuzzy modeling with MFRLS, model optimally scheduling, and multiple-variable general predictive control. The OSMM-GPC algorithm works in the following procedure:

First, the OSMM-GPC uses Takagi–Sugeno fuzzy modeling to build $C(n, m)$ groups of $m \times m$ TS fuzzy models, where $C(n, m)$ is the number of m combinations from the set of n boilers; based on these TS models, $C(n, m)$ groups of adaptive prediction models

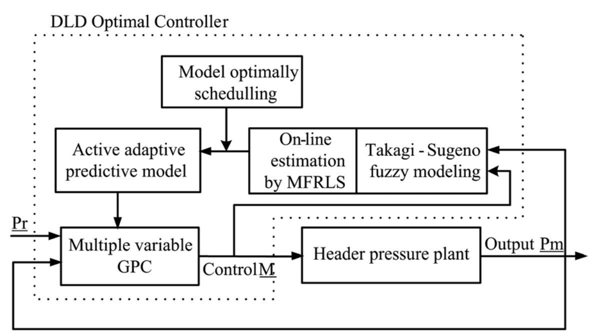

Fig. 2 A DLD optimal control system with OSMM-GPC algorithm

are online estimated by MFRLS algorithm at each sampling interval. Then the model optimally scheduling algorithm chooses the optimal one from the $C(n, m)$-adaptive models on an economic objective as the active adaptive prediction model. It is used by the $m \times m$ general predictive controller to calculate the control increment ΔM of the current sampling instant, where ΔM denotes m-dimension vector of the incremental firing rates. The chosen boilers will add ΔM to their previous firing rates, and the unselected boilers keep theirs. In Fig. 2, M denotes the n-dimension vector of firing rates; P_{r} denotes the m-dimension vector of the set points of the header pressures; P_{m} denotes the m-dimension vector of the header-pressure measurements.

4.1 The Model Optimally Scheduling Algorithm. After the Takagi–Sugeno fuzzy modeling and the online model updating in each sampling interval, we have the novel set of $C(n,m)$-adaptive prediction models

$$\mathrm{PM} = [\mathrm{PM}_1, \cdots, \mathrm{PM}_{C(n,m)}]' \tag{26}$$

Each model in the set has m inputs and m outputs. There are $C(n, m)$-input combinations as the control scheme candidates

$$\Delta M_i = [\Delta M_{p_i(1)}, \cdots, \Delta M_{p_i(m)}]', \quad i = 1, \cdots, C(n,m) \tag{27}$$

where the suffix $p_i(j)$, $j = 1, \ldots, m$, is the number of the selected boiler of the ith scheme in the boiler set. From the identification of the TS fuzzy model, we know that the inertia influences among different outputs are very small. Thus, each m-input-and-m-output model in PM can be approximately described as a group of m-input-and-1-output models like the following [18]:

$$\mathrm{PM}_{i:} \quad \begin{aligned} & A_{i,j}(q^{-1})Y_j(k) = \sum_{l=1}^{m} \left[B_{i,j,l}(q^{-1})\Delta M_{p_i(l)}(k-1)\right] + c_{i,j} \\ & j = 1, \cdots m \quad, i = 1, \cdots, C(n,m) \end{aligned} \tag{28}$$

where q^{-1} is the backward shifting operator, $C_{i,j}$ is the affine item from the linearization modeling. $Y_j(k)$ is the jth output, and

$$A_{i,j}(q^{-1}) = 1 + a_{i,j,1}q^{-1} + \cdots + a_{i,j,n_{i,j}}q^{-n_{i,j}} \tag{29}$$

$$\begin{aligned} B_{i,j,l}(q^{-1}) = {} & b_{i,j,l,0}q^{-d_{i,j,l}} + b_{i,j,l,1}q^{-d_{i,j,l}+1} \\ & + \cdots + b_{i,j,l,(m_{l,i}-1)}q^{-(d_{i,j,l}+m_{i,j,l}-1)} \quad l = 1, \cdots, m, j = 1, \\ & \cdots, m, i = 1, \cdots, C(n,m) \end{aligned} \tag{30}$$

Equations (26)–(30) can represent not only the dynamics of the plant but also the efficiency of each boiler for regulating the pressures at the current operation point. We evaluate the efficiency of each scheme by the Final Value Theorem. Let each input be a step signal, and the steady gain of each model is obtained by equating the operator q^{-1} to 1. Because the affine item $C_{i,j}$ represents an offset from a linearization center of these submodels, it is pointless on evaluating efficiency. Thus, we have the regulating efficiency (MPa/(kg/s))[24]

$$\eta(\mathrm{PM}_i) = \sum_{j=1}^{m} \omega_j \frac{A_{i,j}(1)}{\sum_{l=1}^{m} B_{i,j,l}(1)}, \quad i = 1, \cdots, C(n,m) \tag{31}$$

where $A_{i,j}(1)$ and $B_{i,j,l}(1)$ are obtained by equating the operator q^{-1} to 1 in Eqs. (29) and (30). Their coefficients come from the online estimation by MFRLS. ω_j represents the importance of the jth outputs. Here, ω_j weights the impacts of these outputs to the overall regulating efficiency. In the distributed system of PUUH, $\omega_j = 1$ is suggested because the pressure points along the header have the equal importance. Equation (31) evaluates the regulating efficiency of the $C(n, m)$ schemes at current operation point. The optimization objective of selecting the boiler combination scheme for the most effective regulating is given by

$$\max_{i \in \{1,\cdots,C(n,m)\}} \eta(PM_i) = \sum_{j=1}^{m} \omega_j \frac{A_{s,j}(1)}{\sum_{l=1}^{m} B_{s,j,l}(1)}, s \in \{1, \ldots, C(n,m)\} \tag{32}$$

where suppose that the optimal scheme is the sth. The solution of problem (32) determines which boiler combination scheme should be adopted to regulate the header pressures at the current operation point. If a boiler is in the selected scheme, it adjusts its firing rate at the current time instant according to the instruction from the OSMM-GPC controller; otherwise, it keeps its previous firing rate.

As a result of solving problem (32), an inner prediction model for the general predictive control algorithm is settled according to the selected scheme. The model of the chosen scheme is given by

$$\begin{aligned} \mathrm{PM}_{s:} \Delta M_s = [\Delta M_{p_s(1)}, \cdots, \Delta M_{p_s(m)}]', s \in \{1, \cdots, C_n^m\}, \\ Y(k) = [Y_1(k), \cdots, Y_m(k)], \\ A_{s,j}(q^{-1})Y_j(k) = \sum_{l=1}^{m} \left[B_{s,j,l}(q^{-1})\Delta M_{p_s(l)}(k-1)\right] + c_{s,j},. \\ j = 1, \cdots m, \end{aligned} \tag{33}$$

where PM_s means the model according to the sth input scheme chosen. Then we can use it as the prediction model of the general predictive controller to calculate new control increment.

In each sampling interval, the OSMM-GPC algorithm first schedules the regulating boiler combination on the objective of the highest regulating efficiency at current operation point, and then it solves a regulatory optimization problem by the multiple-variable general predictive control algorithm on the selected boilers.

4.2 Multiple-Variable General Predictive Control. The regulatory optimization problem of PUUH is described as

$$\begin{aligned} \min J = \min \Bigg\{ & \sum_{j=N_1}^{N_2} [Y(k+j) - w(k+j)]' \lambda(j) [Y(k+j) - w(k+j)] \\ & + \sum_{j=1}^{N_u} \Delta M_s(k+j-1)' Q(j) \Delta M_s(k+j-1) \Bigg\} \end{aligned} \tag{34}$$

$$\begin{aligned} s.t. \quad & \Delta M_s = [\Delta M_{p_s(1)}, \cdots, \Delta M_{p_s(m)}]', s \in \{1, \cdots, C_n^m\} \\ & Y(k) = [Y_1(k), \cdots, Y_m(k)]' \quad , \\ & A_{s,j}(q^{-1})Y_j(k) = \sum_{l=1}^{m} \left[B_{s,j,l}(q^{-1})\Delta M_{p_s(l)}(k-1)\right] + c_{s,j} \\ & j = 1, \cdots, m \\ & M_{\min} < M_{P_s(j)} < M_{\max} \\ & \Delta M_{\min} < \Delta M_{P_s(j)} < \Delta M_{\max} \end{aligned} \tag{35}$$

where N_1 is the minimum costing horizon; N_2 is the maximum costing horizon; $M_{\min}$ and $M_{\max}$ are the minimal firing rate and maximal firing rate, respectively; $\Delta M_{\min}$ and $\Delta M_{\max}$ are the minimal increment firing rate and maximal increment firing rate, respectively; $\lambda(j)$ is a output-weighting sequence; $Q(j)$ is a control-weighting sequence. $w(k+j)$ is the reference sequence which is obtained from

$$\begin{cases} Y_r(k) = [Y_{r,1}(k), \cdots, Y_{r,m}(k)]' \\ w(k) = Y_r(k) \\ w(k+j) = aw(k+j-1) + (1-a)w(k) \end{cases} \quad j = 1, 2, \ldots \tag{36}$$

where $Y_r(k)$ is the m-dimension vector of the set points of the outputs; $a \in [0, 1)$ is for smoothing the reference sequence. In the multiple-variable constrained general predictive control, $Y(k+j)$ is the j-step ahead optimal prediction of the output, which is given by

$$Y_i(k+j) = G_j(q^{-1})\Delta M(k+j-1) + F_j(q^{-1})Y_i(k) + H_j(q^{-1})\Delta M(k-1) \quad i = 1,\cdots,m \tag{37}$$

$$E_j(q^{-1})B(q^{-1}) = G_j(q^{-1}) + q^{-j}H_j(q^{-1}), \quad j = 1,2,\ldots \tag{38}$$

where $E_j(q^{-1})$ and $F_j(q^{-1})$ are polynomials uniquely defined given $A_{s,j}(q^{-1})$ and the prediction interval j. They satisfy the identity

$$1 = E_j(q^{-1})A_{s,j}(q^{-1})\cdot\Delta + q^{-j}F_j(q^{-1}), \tag{39}$$

where $\Delta = 1 - q^{-1}$.

This optimization problem including Eqs. (34)–(39) is the description of a multiple-variable constrained general predictive control with an adaptive prediction model. It can be transformed into the expression of a standard quadratic programming problem [25] to get feasible solutions of the firing rate increment ΔM. Then, the solution is added into the m boilers selected by the model optimally scheduling algorithm at the current operation point. The other (n–m) boilers keep their previous firing rates. The architecture of Fig. 1 provides smooth switching of the control increment among the boilers.

Because a low-dimension multiple-variable OSMM-GPC algorithm takes less computing time, the configuration of such a DLD optimal controller with the OSMM-GPC algorithm depends on the size of a PUUH system. If the system is small, one controller with the multiple-variable OSMM-GPC algorithm can meet its needs; if the system is large, then it is more proper to set one DLD controller for each one balanceable group as mentioned in Sec. 2.

5 Simulations and Discussion

In this section, we illustrate OSMM-GPC algorithm on its two novel components of MFRLS and the model optimally scheduling algorithm. For clearly showing the tracking performance of MFRLS, we verify it on a 2-input-and-2-output water tank model which is an identification bench mark [18]. It has small initial dynamics and is sensitive to the noise; thus, it is proper for testing estimation performance. If an estimation approach can track this time-varying model well, it must be able to track other slow time-varying and large-initial plant like PUUH. In the second part, we will illustrate the effectiveness of the model optimally scheduling algorithm on a typical PUUH theoretical model [11] with three boilers and two turbines.

5.1 Verifying MFRLS We illustrate the new MFRLS algorithm using the following nonlinear theoretical model of a 2-input-and-4-output water tank group [5]:

$$\begin{bmatrix}\dot h_1\\ \dot h_2\\ \dot h_3\\ \dot h_4\end{bmatrix} = \begin{bmatrix}-2.82 & 0 & 2.256 & 0.564\\ 0 & -2.82 & 0.564 & 2.256\\ 0 & 0 & -2.82 & 0\\ 0 & 0 & 0 & -2.82\end{bmatrix}\cdot\begin{bmatrix}\sqrt{h_1}\\ \sqrt{h_2}\\ \sqrt{h_3}\\ \sqrt{h_4}\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 15.12 & 0\\ 0 & 15.12\end{bmatrix}\cdot\begin{bmatrix}u_1\\ u_2\end{bmatrix} \tag{40}$$

where h_i, $i = 1, 2, 3, 4$ denotes the water level in the ith tank; u_1 and u_2 denote two input flows; all are dimensionless. Using the Gustafson–Kessel clustering and batch least-squares estimation to identify the antecedent and consequent parameters of its TS model, we have the initial TS model of h_1 and h_2 shown in Table 1.

For verifying the online estimating performance of MFRLS, we simulate a time-varying case of the consequent parameters of the TS model as real changes on the plant in a long-term operation. In the theoretical model (Eq. (40)), h_3 and h_4 are the inputs of h_1 and h_2. Two kinds of variations are made in Eq. (40) after the 500-step simulations: one is the ramp change on coefficients of the inputs and the other is the step change from 1 to 450 s on the input delays. Both start at 500-step. The modifications on Eq. (40) are given by

$$h_1(k) = h_1(k-1) - 2.82\cdot\sqrt{h_1(k-1)} + (1-0.0005\cdot k)\cdot 2.256 \cdot\sqrt{h_3(k-450)} \quad + (1-0.0005\cdot k)\cdot 0.564\cdot\sqrt{h_4(k-450)} \tag{41}$$

$$h_2(k) = h_2(k-1) - 2.82\cdot\sqrt{h_2(k-1)} + (1-0.0005\cdot k)\cdot 0.564 \cdot\sqrt{h_3(k-450)} \quad + (1-0.0005\cdot k)\cdot 2.256\cdot\sqrt{h_4(k-450)} \tag{42}$$

We use the binary pseudo-random sequences as the inputs u_1 and u_2, shown in Fig. 3, to stimulate the theoretical model (40) with above time-varying characteristics. There are major errors between the outputs of the TS model without online estimation and the theoretical model. For comparing the tracking capabilities of each method, we make the online estimations on the consequent coefficients of the TS model of h_1 and h_2 by the local FRLS, the global FRLS, and the MFRLS, respectively. The tracking outputs are shown in Figs. 4(*a*)–4(*c*). The curves composed of "x" represent the outputs of the TS model, and the real curves represent the outputs of the theoretical model. Table 2 shows their mean-absolute errors which are given by

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y(i) - \hat y(i)| \tag{43}$$

Obviously, the local FRLS worked better than the global FRLS in online estimation because it considers more features of the sampling points in the algorithm, but not sufficiently. The tracking performance of MFRLS is better than the other two in all the cases. When the input has a large change which means the state transfers from one subspace into another subspace, MFRLS shows great advantage over the local FRLS. This will bring advantages to properly scheduling and smoothly switching in an optimally scheduling multiple-model control.

5.2 Verifying the Model Optimally Scheduling Algorithm of OSMM-GPC. Because the performance of general predictive control with Takagi–Sugeno model for the long delay and nonlinear plants has been extensively demonstrated both in the literature and practice, the emphasis of this experiment is to verify the effect of the model optimally scheduling algorithm, i.e., to compare the two aspects of the fuel consumptions and the dynamic

Table 1 Rules of TS model

R_1	If ($h_1(k-1)$ and $h_2(k-1)$ and $h_3(k-1)$ and $h_4\,(k-1)$) are in Ω_1 Then $h_1(k) = 0.411h_1(k-1) + 0.653h_3(k-1) + 0.0774h_4(k-1) - 0.0211$ and $h_2(k) = 0.176h_2(k-1) + 0.0843h_3(k-1) + 0.873h_4(k-1) - 0.0049$
R_2	If ($h_1(k-1)$ and $h_2(k-1)$ and $h_3(k-1)$ and $h_4\,(k-1)$) are in Ω_2 Then $h_1(k) = 0.57h_1(k-1) + 0.3h_3(k-1) + 0.176h_4(k-1) - 0.0008$ and $h_2(k) = 0.596h_2(k-1) + 0.108h_3(k-1) + 0.317h_4(k-1) - 0.0256$
R_3	If ($h_1(k-1)$ and $h_2(k-1)$ and $h_3(k-1)$ and $h_4(k-1)$) are in Ω_3 Then $h_1(k) = 0.74\,h_1(k-1) + 0.186\,h_3(k-1) + 0.099\,h_4(k-1) - 0.0091$ and $h_2(k) = 0.717\,h_2(k-1) + 0.118\,h_3(k-1) + 0.175\,h_4(k-1) - 0.0101$

Note: Ω_i, i = 1, 2, 3, is the fuzzy set.

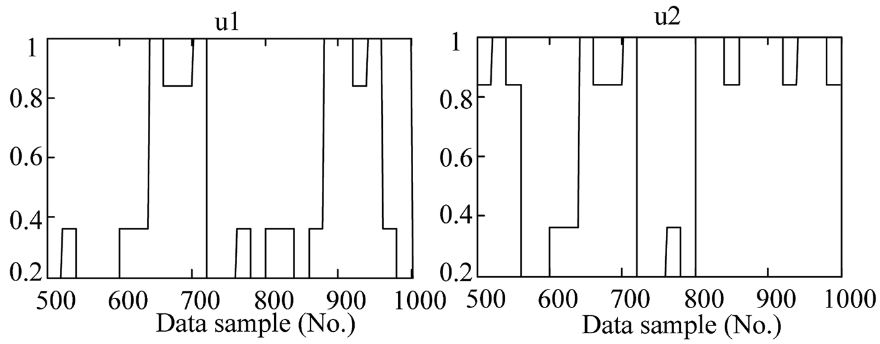

Fig. 3 The inputs of the water levels

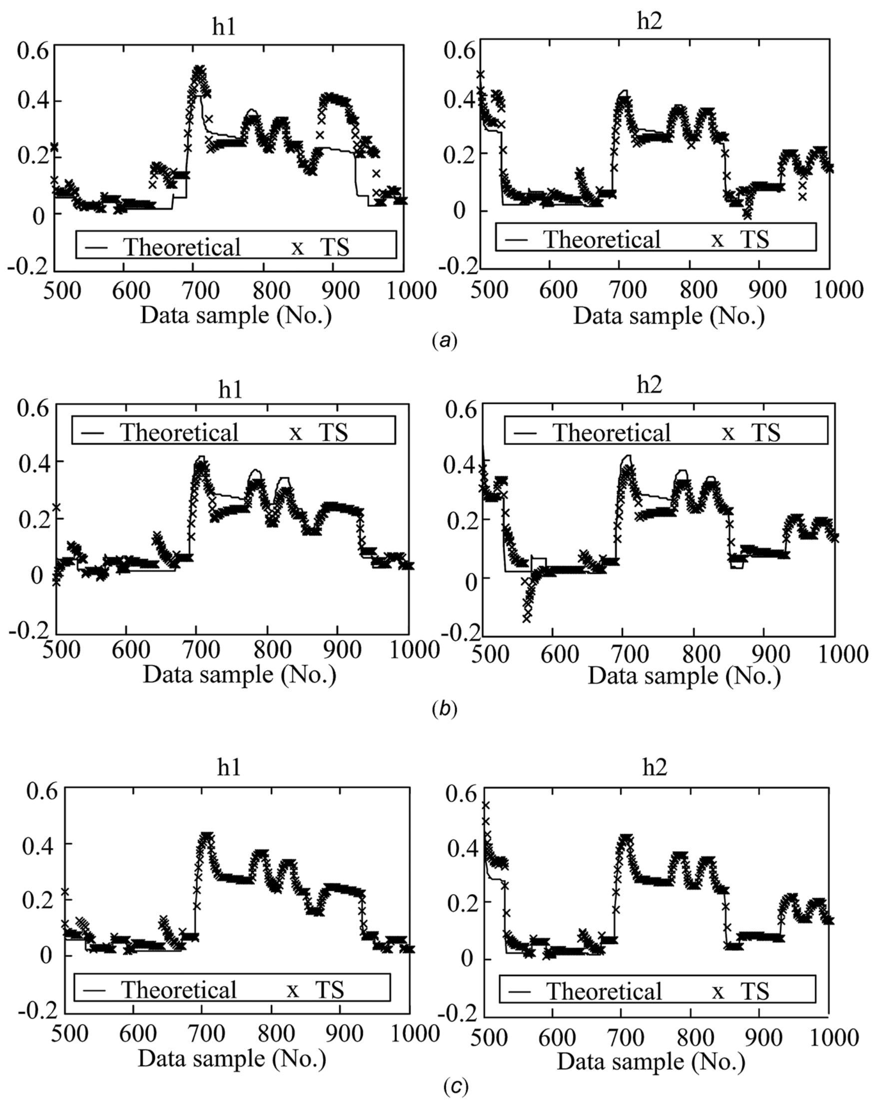

Fig. 4 Online estimated outputs of the water levels: (*a*) global optimal estimation, (*b*) local optimal estimation, and (*c*) mixed optimal estimation

Table 2 MAE of simulations

MAE	h_1	h_2
Global FRLS	0.0523	0.0260
Local FRLS	0.0236	0.0204
MFRLS	0.0174	0.0126

performance between the general predictive controls with and without the optimally scheduling on their prediction models.

We did the experiment on a theoretical model [11] of a typical PUUH system with three boilers and two steam turbines. The full steam flow of each boiler or steam turbine in this system is 61 kg/s. First, we obtained three initial groups of 2-input-and-2-output prediction models by TS modeling. At each sampling interval, the online model identifying the function updated the three groups of prediction models by MFRLS to precisely track the slow time-varying system in practice. Then, the model optimally scheduling algorithm chose one group with the maximal regulating efficiency by Eq. (32) among the three groups of models and made it as the active prediction model of the general predictive controller to calculate the control increment at current sampling instant.

In the experiment, first let the theoretical PUUH model run over 2000 steps to arrive at an initial steady state, where the header pressures, $P_{m2} = P_{m4} = 8.75$ MPa, need to be kept as the set point all along and the opening ratios of the steam turbine valves, $V_1 = V_2 = 50\%$, varied according to the DLD. The DLD can be simulated in experiments through tuning the two throttle valves freely. Thus at the 2050th step, let the opening ratio of the steam turbine valve 1 step from 50% to 70% to simulate the increase of the DLD; after about 700 steps running under the pressure control, the PUUH arrived at a new stable state, then at the 3500th step, let the opening ratio of the steam turbine valve 2 step from 50% to 35% to simulate the decrease of the DLD. These variations of the DLD are shown in Fig. 5(*a*). The three control ways of OSMM-GPC, GPC with controllable boilers 1 and 2, and GPC with controllable boilers 1 and 3 were individually implemented in the experiment.

Integrating the firing rate summation of all the three boilers from the 2000th step to the 4500th step and comparing the fuel consumptions among the three control ways produces the results shown in Table 3. Obviously, OSMM-GPC consumed the least fuel among the three ways in tracking the same DLD trend of Fig. 5(*a*). The saved fuel is about 8199 t for 1 year. "①" in Fig. 5(*d*) points out the switching of the manipulations between boilers 2 and 3. Because boiler 3 had higher efficiency than boiler 2 in the range before that point, the two manipulated variables of the incremental firing rates at each sampling instant were sent into boilers 1 and 3, meanwhile, boiler 2 kept its constant firing rate; in the range after that point, the incremental firing rates were sent into boilers 1 and 2 because boiler 2 had higher efficiency than boiler 3 then, meanwhile, boiler 3 retained its value. Boiler 1 had the highest efficiency among the three all along. As a result of efficient control, the OSMM-GPC shows much better transient performance on P_{m2} and P_{m4} than the other two ways, i.e., both the transient curves of P_{m2} in Fig. 5(*b*) and P_{m4} in Fig. 5(*c*) show that OSMM-GPC control had less overshoot and shorter regulating time than the other two ways in tracking the same DLD trend of Fig. 5(*a*). Figures 5(*e*) and 5(*f*) show the firing rates of the fixed patterns of GPC.

Therefore, we draw a conclusion that the OSMM-GPC can not only improve the DLD following capability of PUUH on both the stability and rapidity of the header-pressure control but also improve the economic profits by saving considerable fuel consumption in its regulating.

Table 3 Comparison of fuel consumptions

Items Approaches	Fuel consumption (kg/2500 s)	Firing rate (kg/s)
GPC with boilers 1 and 2	40,525	16.21
GPC with boilers 1 and 3	40,475	16.19
OSMM-GPC	39,825	15.93

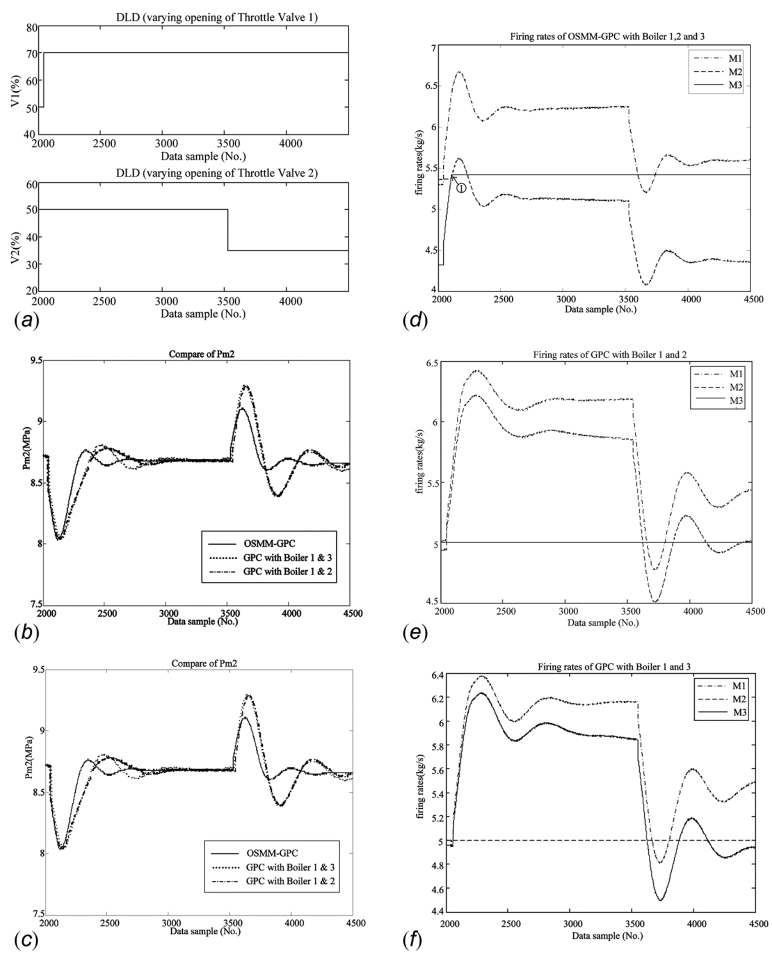

Fig. 5 Pressure control performance comparison between the OSMM-GPC and the standard GPC while tracking DLD

6 Conclusions

For improving DLD-following capability and the economic profit of a PUUH system, this paper has proposed a distributed multiple-model general predictive control algorithm OSMM-GPC. The OSMM-GPC is built on a distributed increment-control architecture which is proper for flexibly scheduling and smoothly switching on devices and models. As a comprehensive control algorithm, the OSMM-GPC has an improved online parameter-estimation algorithm MFRLS and a model optimally scheduling algorithm. The MFRLS algorithm can provide precise parameter estimation on the TS fuzzy model of a time-varying system. It sufficiently considers the membership feature of each sampling point to mix the local estimation and global estimation together by making a dynamical multi-objective cost function. With sufficient use of the information from sampling data, MFRLS has a better approximation to the real system than the global and local FRLS. It is a general parameter-estimation approach for building adaptive models in wide applications.

Based on the high-precision adaptive models, an optimal scheduling of control devices and their models can be well executed. The model optimally scheduling algorithm computes the regulating efficiencies of all control groups on their adaptive models and then chooses the optimal one to accept the control signals from a multiple-variable general predictive controller. Through the scheduling at each operation point, considerable fuel consumption can be saved in regulating the header pressures or tracking DLD, meanwhile, the control performance indexes including overshoots and regulating time are much better than that of the control without model optimally scheduling as proved in the simulating experiment.

As a comprehensive control algorithm, the OSMM-GPC is not only able to improve the control performance of header pressures, the DLD-following capability, and the economic profit of PUUH but also can work for a wide range of industry processes which need distributed multiple-model control.

Acknowledgment

The authors are grateful to the support of the National Natural Science Foundation of China under Grant Nos. 51106024 and 51036002. The authors also sincerely thank Dr. Xinsheng Lou for many of his suggestions and encouragements.

References

[1] Pan, L., and Shen, J., 2010, "A Hierarchical Economical Load Dispatch Based on Flow-Balanced Zones of Parallel Coursing Power Units With a Header," *Proceedings 53rd ISA POWID Symposium*, Summerlin, NV, Vol. 483, pp. 410–418.

[2] Warsono King, D. J., Ozveren, C. S., and Bradley, D. A., 2007, "Economic Load Dispatch Optimization of Renewable Energy in Power System Using Genetic Algorithm," IEEE Proc. Power Tech. Lausanne, **1**, pp. 2174–2179.

[3] Chiou, P. J., 2009, "A Variable Scaling Hybrid Differential Evolution for Solving Large-Scale Power Dispatch Problems," IET Gener. Transm. Distrib., **3**(2), pp. 154–163.

[4] Majanne, Y., 2005, "Model Predictive Pressure Control of Steam Networks," Control Eng. Pract., **13**, pp. 1499–1505.

[5] Luyben, W. L., 2004, "Mathematical modeling and control of a multiboiler steam generation system," Ind. Eng. Chem. Res., **43**(8), pp. 1839–1852.

[6] White, D., 1999, "Robust control of a steam header letdown system," IEEE Canadian Conference on Electrical and Computer Engineering, **2**, pp. 937–940.

[7] Emoto, G., Tsuda, A., Takeshita, T., Monical, M. T., Nakagawa, S., and Fujita, K., 1998, "Integrated Large-Scale Multivariable Control and Real-Time Optimization of a Power Plant," IEEE Proc. Control Applications, **2**, pp. 1368–1372.

[8] Clarke, D. W., Mohtadi, C., and Tuffs, P. S., 1987, "General Predictive Control-Part II. Extension and Interpretations," Automatica, **23**(2), pp. 149–160.

[9] Clarke, D. W., Mosca, E., and Scattolini, R., 1994, "Robustness of an Adaptive Predictive Controller," IEEE Trans. Autom. Control, **39**(5), pp. 1052–1056.

[10] Liu, X. J., Guan, P., and Chan, C. W., 2010, "Nonlinear Multivariable Power Plant Coordinate Control by Constrained Predictive Scheme," IEEE Tran. Control Syst. Technol., **18**(5), pp. 1116–1125.

[11] Pan, L., and Shen, J., 2010, "Study of Approximate Distributed Dynamic Model of Multisource and Multi-Sink Steam Manifold System of Thermal Power Plant," Chem. Eng. Commun., **197**(2), pp. 204–212.

[12] Li, N., Li, S. Y., and Xi, Y. G., 2004, "Multi-Model Predictive Control Based on the Takagi-Sugeno Fuzzy Models: A Case Study," Inf. Sci., **165**, pp. 247–263.

[13] Narendra, K. S., Balakrisham, J., and Ciliz, M. K., 1995, "Adaption and Learning Using Multiple Models, Switching and Tuning," IEEE Control Syst. Mag., **15**(3), pp. 37–51.

[14] Prasad, G., Swidenbank, E., and Hogg, B. W., 1998, "A Local Model Networks Based Multivariable Long-Range Predictive Control Strategy for Thermal Power Plants," Automatica, **34**(10), pp. 1185–1204.

[15] Habbi, H., Zelmat, M., and Bouamama, B. O., 2003, "A Dynamic Fuzzy Model for a Drum-Boiler-Turbine System," Automatica, **39**(7), pp. 1213–1219.

[16] Kallapa, P., and Ray, A., 2000, "Fuzzy Wide-Range Control of Fossil Power Plants for Life Extension and Robust Performance," Automatica, **36**(1), pp. 69–82.

[17] Pan, L., Shen, J., and Luh, P. B., 2010, "A Mixed Fuzzy Recursive Least-Squares Estimation for Online Identification of Takagi-Sugeno Models," *Proceedings IEEE Conference on Progress in Informatics and Computing*, Shanghai, CN, Vol. 1, pp. 326–330.

[18] Babuška, R., and Verbruggen, H. B., 1998, "Identification of MIMO System by Input-Output TS Fuzzy Models," *Proceedings FUZZ-IEEE*, Anchorage, AK, Vol. 1, pp. 657–662.

[19] Abonyi, J., Babuška, R., and Szeifert, F., 2004, "Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., **32**(10), pp. 612–621.

[20] Gath, L., and Geva, A., 1989, "Unsupervised Optimal Fuzzy Clustering," IEEE Trans. Pattern Anal. Mach. Intell., **7**, pp. 773–781.

[21] Hadjili, M. L., and Werta, V., 2002, "Takagi-Sugeno Fuzzy Modeling Incorporating Input Variables Selection," IEEE Trans. Fuzzy Syst., **10**(6), pp. 728–742.

[22] Yen, J., and Wang, L., 1998, "Improving the Interpretability of TSK Fuzzy Models by Combined Local and Global Learning," IEEE Trans. Fuzzy Syst., **6**(4), pp. 530–537.

[23] Angelov, P. P., and Filev, D. P., 2004, "An Approach to Online Identification of Takagi-Sugeno Fuzzy Models," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., **34**(1), pp. 484–498.

[24] Johns, A. T., and Warne, D. F., 2003, *IEE Power and Energy Series: Thermal Power Plant Simulation and Control*, The Institution of Electrical Engineers, UK, Vol. 43, pp. 172–175.

[25] Dimitri, P. B., 1999, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA.