Iterative Learning Double Closed-Loop Structure for Modeling and Controller Design of Output Stochastic Distribution Control Systems

Jinglin Zhou, Hong Yue, Jinfang Zhang, and Hong Wang, *Senior Member, IEEE*

Abstract—Stochastic distribution control (SDC) systems are known to have the two-dimensional characteristics regarding time and probability space of a random variables, respectively. A double closed-loop structure, which includes iterative learning (IL) modeling (ILM) and iterative learning control (ILC), is proposed for non-Gaussian SDC systems. The ILM is arranged in the outer loop, which takes a longer period for each cycle termed as a *BATCH*. Each *BATCH* is divided into a modeling period and a number of control intervals, called *batch*es, being arranged in the inner loop for ILC. The output probability density functions (PDFs) of the system are approximated by a radial basis function neural network (RBFNN) model, whose parameters are updated via ILM in each *BATCH*. Based on the RBFNN approximation of the output PDF, a state-space model is constructed by employing the subspace parameter estimation method. An IL optimal controller is then designed by decreasing the PDF tracking errors from *batch* to *batch*. Model simulations are carried out on an 4th-order numerical example to examine the effectiveness of the proposed algorithm. To further assess its application feasibility, a flame shape distribution control simulation platform for a combustion process in a coal-fired gate boiler system is constructed by integrating WinCC interface, Matlab simulation programs and OPC communication together. Simulation study over this industrial simulation platform shows that, the output PDF tracking performance can be efficiently achieved by this double closed-loop IL strategy.

Index Terms—Iterative learning (IL), optimal tracking control, probability density function (PDF), radial basis function neural network (RBFNN), stochastic distribution control (SDC), subspace identification, temperature field distribution control.

I. INTRODUCTION

Stochastic control has been one of the most energetic areas in control theory and applications for the simple fact that many real industrial processes are subject to various sources of randomness. Following the development of minimum variance control [1] and linear quadratic (LQ) control [2], a number of approaches have been developed and applied in real applications such as upper-bound minimum variance control, stochastic adaptive control, stochastic optimization and forecasting, sliding mode control [3]–[5], etc. In these methods, mean and variance are considered in controller design for systems with Gaussian random signals. For systems with non-Gaussian signals such as food processing unit, jet flame control in furnace systems, molecular weight distribution control (MWD) or particle size distribution (PSD) control in polymerization processes [6]–[8], mean and variance control can be largely limited and higher-order distribution information is required. To address this problem, a new method which directly controls the distribution function of system output is developed, which is referred to as stochastic distribution control (SDC) or probability density function (PDF) shaping control [9]–[11].

The output PDF modeling and control was originally proposed to solve the distribution control problems in paper making machines where the paper sheet produced needs to have uniform distribution in both longitude direction and cross direction [9]. In this method, there are two subspaces regarding the time sequence and the probability distribution, respectively. Further development included optimal PDF tracking control, robust PDF tracking control and minimum entropy control, etc. [12]–[19], which brings improvement in PDF control from various aspects.

To establish a dynamic PDF model for the purpose of closed-loop control, B-spline neural networks are normally used to approximate the output PDF where the basis function weights are calculated from the PDF fitting, and the model for weights dynamics can be obtained by least-square estimation or other regression methods. Fixed B-spline basis functions are used in most PDF modeling. To be adaptive to large process variations, more recently, a iterative learning control (ILC) strategy has been presented using a fixed structure controller for PDF control while iteratively tuning the basis function parameters, thus the basis functions, for PDF approximation [14], [22], [23]. In fact, the ILC in output SDC systems can be realized by iteratively adjusting the controller and/or updating the basis functions in PDF approximation. In the former case, the ILC design is implemented to tune the controller while keeping the basis functions fixed. Control algorithms developed in this way take two assumptions, one is that the output PDFs can be well approximated by the chosen (fixed) basis functions, another is that the dynamic characteristics of the developed model is reasonably close to that of the actual system. In the case of IL for model update, the parameters

J. Zhou is with College of Information Science and Technology, Beijing University of Chemical Technology, Beijing P.R. China (e-mail: jinglinzhou@mail.buct.edu.cn).

H. Yue is with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, Scotland, UK (e-mail: hong.yue@strath.ac.uk).

J. Zhang is with the School of Automation and Computer Engineering, North China Electric Power University, Beijing 102206, P.R. China (e-mail: jfzhang@ncepu.edu.cn).

H. Wang is with the Control Systems Centre, University of Manchester, Manchester M60 1QD, UK (e-mail: hong.wang@manchester.ac.uk).

of the basis functions in PDF approximation are iteratively adjusted while keeping a fixed controller. The assumption in the latter case is that the fixed controller will satisfy the control requirements during the period of model updating, which is not always easy to guarantee since the change of modeling will inevitably affect the control performance.

ILC is a model-free controller, but the learning rate of ILC can be better selected and adjusted when a system model is available. In general, it is difficult to measure or estimate the output PDF for SDC systems in an online fashion. This makes it hard to achieve online closed-loop control of output PDF. In ILC of a SDC system, the quality of the model affects not only the online estimation of the output PDF, but also the learning rate adjustment in controller design. A good modeling relies on the choice of informative data, and also the radial basis function neural network (RBFNN) for the output PDFs approximation. The RBFNN describes the relationship between the parameters of the basis functions and the output PDF. The modeling error in PDF approximation can be reduced by adjusting the parameters of the basis functions properly. Therefore, in order to improve the approximation accuracy of a PDF model and the control performance, it is necessary to update the dynamic PDF model following the adjustment of the basis functions.

In a standard ILC, the control horizon is divided into a number of batches normally of the same length in time. When the modeling and controller design are considered together, the ILC with a fixed controller structure may have two disadvantages. One is that both the model parameter identification and the PDF tracking controller design are performed within each batch, which not only involves intensive computational efforts but also causes frequent change of model parameters. The other drawback is that the controller update is only related to the tuning of the basis functions, which could be inadequate due to the integral action on the control performance index. In this work, an IL-based double closed-loop structure for modeling and controller design is proposed for SDC systems where the whole control process is divided into several large modeling periods, each one called a *BATCH*, and each *BATCH* is divided into a number of control periods, called *batch*es. This new structure achieves two objectives: (1) in each *BATCH*, the basis function parameters are tuned to adjust the shape of the basis functions used in PDF approximation, and then a state-space model is established via subspace parameter estimation using the updated B-splines; (2) an optimal ILC is designed within each batch to make the system output PDF follow the target PDF in a timely manner.

The remaining part of this article is organized as follows. The output PDF model representation based on the RBFNN and subspace identification are briefly introduced in Section II. The optimal ILC algorithm and the RBF parameter tuning method are given in Section III. In Section IV, the double closed-loop structure and the associated controller design and model identification are developed. In Section V, the effectiveness of the proposed method is discussed through numerical simulations. The application study is carried out in Section VI where a simulation platform is established for temperature field distribution control operation in a chain of grate boilers and this platform is employed for the simulation study of the double closed-loop PDF control. Conclusions are given in Section VII.

II. PROBLEM FORMULATION

A. Model Representation

For a dynamic stochastic system, define $o(t) \in [a, b]$ as a uniformly bounded random variable for the output of the system and $u(t) \in \mathbb{R}^{d \times 1}$ as the input vector to control the distribution shape of the output. At any time t, the distribution shape of $o(t)$ can be represented by its PDF $\gamma(y, u(t))$ [9],

$$P\{a \le y \le \zeta, u(t)\} = \int_a^\zeta \gamma(y, u(t)) dy \qquad (1)$$

where $P\{a \le y \le \zeta, u(t)\}$ represents the probability of the random variable y locating in $[a, \zeta]$ with control $u(t)$. Similar to [9], the output PDF is approximated by a truncated Gaussian type of RBFNN:

$$R_l(y) = \exp\left(-\frac{(y - \mu_l)^2}{\sigma_l^2}\right), \ l = 1, 2, \cdots, n \qquad (2)$$

where l is the index of the network nodes and n is the total number of nodes; μ_l and σ_l are the parameters representing the center and width of the node function $R_l(y)$. The output PDF can be approximated by this RBFNN as

$$\gamma(y, u_k) = \sum_{l=1}^n w_{l,k} R_l(y) + e_k(y) \qquad (3)$$

where the subscript k is the sample time index, $w_{l,k}$ is the weight of the l-th node , $e_k(y)$ represents the PDF approximation error at time k and $\int e_k(y) dy = 0$. Define $r_l = \int_a^b R_l(y) dy$, $\lambda_l = R_l(y) - R_n(y) r_l / r_n$. Let $L(y) = r_n^{-1} R_n(y)$, $\Lambda(y) = [\lambda_1, \cdots, \lambda_{n-1}]$, and $V_k = [w_{1,k}, \cdots, w_{n-1,k}]^{\mathrm{T}}$, the PDF approximation in (3) can be written as

$$\gamma(y, u_k) = \Lambda(y) V_k + L(y) + e_k(y). \qquad (4)$$

Ignoring $e_k(y)$ for simplicity, the RBFNN approximate weights vector, V_k, can be computed from the following equation

$$V_k = \left[\int_a^b \Lambda^{\mathrm{T}}(y) \Lambda(y) dy\right]^{-1} \int_a^b \Lambda^{\mathrm{T}}(y) [\gamma(y, u_k) - L(y)] dy. \qquad (5)$$

Equation (5) shows the relationship between the output PDF $\gamma(y, u_k)$ and the approximation weights vector V_k. Considering linear dynamics in V_k, a standard state-space model can be written for the output SDC systems:

$$
\begin{aligned}
x_{k+1} &= A x_k + B u_k \\
V_k &= C x_k + D u_k \\
\gamma(y, u_k) &= \Lambda(y) V_k + L(y).
\end{aligned}
\qquad (6)
$$

where $x_k \in \mathbb{R}^{q \times 1}$ is the process state vector, $u_k \in \mathbb{R}^{d \times 1}$ is the control input vector, $V_k \in \mathbb{R}^{p \times 1}$ is the approximation weights vector ($p = n - 1$ from (3)). $A \in \mathbb{R}^{q \times q}$, $B \in \mathbb{R}^{q \times d}$, $C \in \mathbb{R}^{p \times q}$ and $D \in \mathbb{R}^{p \times d}$ are parameter matrices. The first and second equations in (6) reveal the state dynamics and

weights dynamics, respectively; the third one is the RBFNN approximation of the output PDF. It's not a trivial task to establish this state-space model using input-output data to meet the required modeling and accuracy with a reasonable computational load. In this work, the subspace identification method is employed to conduct online parameter estimation, the efficiency of which is higher than other commonly used parameter estimation methods such as the recursive least square (RLS) estimation. In the subspace identification, the main computing load is in matrix calculations. The computing speed won't be much affected when the order of the system is increased. It is therefore also suitable for online estimation of multivariable systems.

Remark 1: If there's no feed-forward loop in the SDC system, model (6) can be simplified as

$$EV_{k+1} = \bar{C}V_k + \bar{D}u_k$$
$$\gamma(y, u_k) = \Lambda(y)V_k + L(y)$$
(7)

where $E = (C^{\mathrm{T}}C)^{-1} C^{\mathrm{T}}, \bar{C} = A (C^{\mathrm{T}}C)^{-1} C^{\mathrm{T}}, \bar{D} = B$.

B. Subspace Modeling

An online estimation of the state-space model in (6) can be achieved through standard subspace identification approaches, for example, N4SID and MOESP [24], [25], to obtain the parameters in (A, B, C, D). The major steps in using MOESP for estimation of (A, B, C, D) is briefed in the following. Further information can be found from literature [24], [25].

Using subspace identification, the Hankel Matrices need to be constructed with the input and output data. Denote the collected input and output data as

$$\begin{pmatrix} u_0 & u_1 & \cdots & u_{s+N-2} \end{pmatrix}$$
$$\begin{pmatrix} V_0 & V_1 & \cdots & V_{s+N-2} \end{pmatrix}$$

respectively, the input and output Hankel matrices are

$$U_{0|s-1} = \begin{bmatrix} u_0 & u_1 & \cdots & u_{N-1} \\ u_1 & u_2 & \cdots & u_N \\ \vdots & \vdots & \ddots & \vdots \\ u_{s-1} & u_s & \cdots & u_{s+N-2} \end{bmatrix} \in \mathbb{R}^{s \times N}$$

and

$$V_{0|s-1} = \begin{bmatrix} V_0 & V_1 & \cdots & V_{N-1} \\ V_1 & V_2 & \cdots & V_N \\ \vdots & \vdots & \ddots & \vdots \\ V_{s-1} & V_s & \cdots & V_{s+N-2} \end{bmatrix} \in \mathbb{R}^{s \times N}$$

where s and N stand for the number of rows and columns of the Hankel matrices, $s + N - 1$ is the total number of samples in time. To guarantee the reliability of the identification, N should be large enough to cover the full dynamic process and s should be larger than the system dimension.

It should be noted that the modeling in this paper is used to facilitate the controller design. To ensure tha all the independent weights can be controlled, theoretically, the minimum order of the model should be $n - 1$. However, the model order should not be too high and the maximum order of the model is recommended to be n for a reliable parameter identification.

The RBFNN-based SDC modeling procedures can be grouped into three steps.
(1) Determine the basis functions. Choose a proper RBFNN to approximate the output PDF.
(2) Collect the modeling data u_k and $\gamma(y, u_k)$ at all sampling points. Use the chosen RBFNN to approximate the output PDFs, the weight vector V_k can be calculated using (5).
(3) Use V_k and the system inputs u_k to construct the input and output data pairs for subspace identification. Implement the steps in MOESP [24], [25] to estimate parameter matrices (A, B, C, D).

III. THE ILC CONTROLLER DESIGN AND THE PARAMETERS UPDATE OF RBFNN

A. The IL-based Controller Design

Following the idea of ILC, divide the control horizon into a number of intervals of the same length in time domain, the first two state-space equations in (6) can be re-written as follows:

$$\begin{cases} x_{j,k+1} = Ax_{j,k} + Bu_{j,k} \\ V_{j,k} = Cx_{j,k} + Du_{j,k} \end{cases}$$
(8)

where j indicates the j-th iteration for controller design, also called *batch* in this paper. k represents the k-th sample time in each *batch*. The following algorithm is formulated similar to the ILC-PDF control strategy in [28], [29]. Only key steps of this algorithm are given in the following without expanding the details.

Combining the two equations in (8) gives the weights vector at the k-th sampling time.

$$V_{j,k} = CA^k x_{j,0} + \sum_{s=0}^{k-1} CA^{k-1-s} Bu_{j,s} + Du_{j,k}$$
(9)

The above equation contains Markov parameters of the system, which can be denoted as:

$$h_0 = D$$
$$h_k = CA^{k-1}B$$

In the j-th control iteration, denote $\mathbb{U}_j$ to group control inputs at all sampling time, $\mathbb{V}_j$ as the corresponding output weights vector with initial condition of $\mathbb{V}_{j,0}$, i.e.,

$$\mathbb{U}_j = [u_{j,0}^{\mathrm{T}}, u_{j,1}^{\mathrm{T}}, \cdots, u_{j,k}^{\mathrm{T}}, \cdots, u_{j,L}^{\mathrm{T}}]^{\mathrm{T}}$$
$$\mathbb{V}_j = [V_{j,0}^{\mathrm{T}}, V_{j,1}^{\mathrm{T}}, \cdots, V_{j,k}^{\mathrm{T}}, \cdots, V_{j,L}^{\mathrm{T}}]^{\mathrm{T}}$$
$$\mathbb{V}_{j,0} = [C^{\mathrm{T}}, A^{\mathrm{T}}C^{\mathrm{T}}, \cdots, (A^k)^{\mathrm{T}}C^{\mathrm{T}}, \cdots, (A^L)^{\mathrm{T}}C^{\mathrm{T}}]^{\mathrm{T}}x_{j,0}$$

L is the total number of sampling points (in the j-th iteration) in the time horizon for controller design. Define G with (A, B, C, D) as follows

$$G = \begin{bmatrix} h_0 & & & & \\ h_1 & h_0 & & \mathbf{0} & \\ h_2 & h_1 & h_0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ h_L & h_{L-1} & \cdots & h_1 & h_0 \end{bmatrix}.$$

The weights vector in (9) collected at all sampling time points can be written into a compact format:

$$\mathbb{V}_j = \mathbb{V}_{j,0} + G\mathbb{U}_j. \tag{10}$$

In the ILC design process, the initial values of $\mathbb{V}_{j,0}$ are kept the same at the beginning of each *batch*.

The following quadratic function is defined to evaluate the ILC performance

$$\begin{aligned} J_{j+1} &= \|\mathbb{E}_{j+1}\|_Q^2 + \|\mathbb{U}_{j+1} - \mathbb{U}_j\|_R^2 \\ &= \mathbb{E}_{j+1}^{\mathrm{T}} Q \mathbb{E}_{j+1} + (\mathbb{U}_{j+1}^{\mathrm{T}} - \mathbb{U}_j^{\mathrm{T}}) R (\mathbb{U}_{j+1} - \mathbb{U}_j) \end{aligned} \tag{11}$$

in which Q and R are the pre-defined positive matrices, $\mathbb{E}_j$ is the output error vector in the j-th *batch* defined by

$$\mathbb{E}_j = \left[V_{j,0}^{\mathrm{T}} - V_g^{\mathrm{T}}, V_{j,1}^{\mathrm{T}} - V_g^{\mathrm{T}}, \cdots, V_{j,L}^{\mathrm{T}} - V_g^{\mathrm{T}} \right]^{\mathrm{T}} \tag{12}$$

The first term in (11) represents the difference between $\mathbb{V}_j$ and the target PDF weights vector while the second term is regarding the control input $\mathbb{U}_j$. From (10) and (12), the output error of the $(j+1)$-th *batch* can be calculated as

$$\mathbb{E}_{j+1} = \mathbb{E}_j + G(\mathbb{U}_{j+1} - \mathbb{U}_j) \tag{13}$$

By substituting (13) into (11), the ILC performance index can be re-written as

$$\begin{aligned} J_{j+1} =& (\mathbb{E}_j + G(\mathbb{U}_{j+1} - \mathbb{U}_j))^{\mathrm{T}} Q (\mathbb{E}_j + G(\mathbb{U}_{j+1} - \mathbb{U}_j)) \\ &+ (\mathbb{U}_{j+1} - \mathbb{U}_j^{\mathrm{T}}) R (\mathbb{U}_{j+1} - \mathbb{U}_j) \end{aligned} \tag{14}$$

Taking partial derivative of (14) with respect to $\mathbb{U}_{j+1}$ gives

$$\frac{\partial J_{j+1}}{2\partial \mathbb{U}_{j+1}} = G^{\mathrm{T}} Q \mathbb{E}_j + [G^{\mathrm{T}} Q G + R](\mathbb{U}_{j+1} - \mathbb{U}_j). \tag{15}$$

An optimal solution will be obtained when $\frac{\partial J_{j+1}}{2\partial \mathbb{U}_{j+1}} = 0$, that is,

$$\mathbb{U}_{j+1} = \mathbb{U}_j - [G^{\mathrm{T}} Q G + R]^{-1} G^{\mathrm{T}} Q \mathbb{E}_j. \tag{16}$$

Denote the control learning rate as $\Gamma = -[G^{\mathrm{T}} Q G + R]^{-1} G^{\mathrm{T}} Q$, the following ILC strategy

$$\mathbb{U}_{j+1} = \mathbb{U}_j + \Gamma \mathbb{E}_j \tag{17}$$

is similar to a P-type learning rule of ILC where G is the correlation matrix of the system Markov parameters. This ILC is terminated when the tracking error performance $\|\mathbb{E}_j\|_Q^2$ is below the given threshold, otherwise it goes to the $(j+1)$-th *batch*.

B. The IL-based Model Update of RBFNN

Apart from the IL operation in controller design, the IL strategy can also be introduced into the update of the RBFNN used for output PDF approximation [14], [22], [23]. In the RBFNN PDF modeling as briefed in Section II, each RBF is determined by two parameters, the center μ_l and the width σ_l. To make an accurate approximation of the output PDF, it's important to set up these two parameters properly for each RBF and also choose the right number of basis functions to construct the RBFNN. The PDF approximation errors need to be kept small, otherwise the poor modeling may deteriorate the controller design. The RBFNN tuning can be dealt with either by adjustment of RBF parameters, and/or by changing the nodes number in the RBFNN. In this paper, the number of nodes is fixed, only the tuning of the two RBF parameters is conducted since they are the main factors determining the PDF approximation accuracy. The RBFNN model update is obtained through the following IL method.

Following the idea of iterative learning, divide the whole control horizon into a number of intervals as *BATCH*es. For an unknown controlled plant, only the control inputs $u_{i,k}$ and the output PDFs $\gamma(y, u_{i,k})$ can be provided, where i indicates the i-th *BATCH*, k represents the k-th (time) sampling point in the data sequence. Consider K samples collected in the modeling period and these data are used M times for the update of RBFNN in each modeling *BATCH*. Using subscript m to stand for the m-th update of RBFNN ($m = 1, \cdots, M$), in the i-th *BATCH* after the m-th tuning of RBFNN, the output PDF calculated from the RBFNN model is $\gamma_{i,m,k}$ at each sample k. The error between the model output PDF and the measured PDF can be defined as

$$\epsilon_{i,m,k}(y) = \gamma_{i,O,k}(y) - \gamma_{i,m,k}(y) \tag{18}$$

where $\gamma_{i,O,k}(y)$ is the 'ideal' or measured PDF at each sample k in the i-th *BATCH*, and $\gamma_{i,m,k} := \sum_{l=1}^n \omega_{l,i,m,k} R_{l,i,m}(y)$, $R_{l,i,m}$ is the l-th RBF in the i-th *BATCH* after the m-th tuning; $\omega_{l,i,m,k}$ can be computed by (5). Due to the complex nonlinear nature of $\epsilon_{i,m,k}$ regarding $\mu_{l,i,m}$ and $\sigma_{l,i,m}$ in (18), it will be too complicated to update the RBFNN model using this error formulation directly. An adaptive algorithm is proposed in the following.

A quadratic performance index is chosen to measure the errors between the 'ideal' or measured PDF and the modeling PDF at each sample k:

$$\bar{J}_{i,m,k} = \int_b^a \epsilon_{i,m,k}^2(y) dy. \tag{19}$$

The measure of modeling errors in the m-th tuning of the i-th *BATCH* is then represented by a vector as

$$\bar{\mathbb{E}}_{i,m} = \left[\bar{J}_{i,m,1}, \bar{J}_{i,m,2}, \cdots, \bar{J}_{i,m,K} \right]^{\mathrm{T}}. \tag{20}$$

The total modeling errors in the i-th *BATCH* after the m-th tuning is given by

$$\bar{\mathrm{E}}_{i,m} = \sum_{k=1}^K \bar{J}_{i,m,k}. \tag{21}$$

The adaptive tuning of the RBFNN parameters should guarantee the decrease of $\bar{\mathrm{E}}_{i,m}$ in the modeling iterations. This can be achieved when the following conditions are satisfied:

$$\begin{aligned} \bar{\mathrm{E}}_{i,m+1} &\le \bar{\mathrm{E}}_{i,m} \\ \bar{\mathrm{E}}_{i+1,1} &= \bar{\mathrm{E}}_{i,M}. \end{aligned} \tag{22}$$

$\bar{\mathrm{E}}_{i+1,1} = \bar{\mathrm{E}}_{i,M}$ means that the initial setting of RBFNN parameters in the $(i + 1)$-th *BATCH* is the same as the last tuning results in the i-th *BATCH*. That is, the modeling error in each *BATCH* is no larger than that in the previous *BATCH*. The termination condition of this ILM can be set with a fixed number of iterations or through a modeling error threshold.

For simplicity, a fixed iteration number (M) is used in this section.

Remark 2: $\bar{\mathbb{E}}_{i,m}$ is the measure of the overall approximation errors in the m-th update in the i-th modeling interval. Denote $\Delta \bar{J}_{i,m,k} = \bar{J}_{i,m,k} - \bar{J}_{i,m-1,k}$. If $\Delta \bar{J}_{i,m,k} \leq 0$, the convergence condition (22) will be held since $\bar{J}_{i,m,k}$ is a non-negative function. An adaptive tuning law should thus be developed aiming to assure $\Delta \bar{J}_{i,m,k} \leq 0$.

Define the increments of the two RBF parameters for $R_{l,i,m}$ as $\Delta\mu_{l,i,m}$ and $\Delta\sigma_{l,i,m}$, respectively. The following adaptive law is applied to the tuning of the RBF parameters:

$$\mu_{l,i,m} = \mu_{l,i,m-1} + \Delta\mu_{l,i,m-1} := \mu_{l,i,m-1} + \Lambda_{\mu,l,i,m}\bar{\mathbb{E}}_{i,m-1}$$
$$\sigma_{l,i,m} = \sigma_{l,i,m-1} + \Delta\sigma_{l,i,m-1} := \sigma_{l,i,m-1} + \Lambda_{\sigma,l,i,m}\bar{\mathbb{E}}_{i,m-1}$$

$$(23)$$

where

$$\Lambda_{\mu,l,i,m} = \varepsilon_{\mu,l,i,m} \cdot \lambda_i \cdot \vec{1}, \quad \Lambda_{\sigma,l,i,m} = \varepsilon_{\sigma,l,i,m} \cdot \lambda_i \cdot \vec{1}$$

in which $\vec{1} \in \mathbb{R}^{1 \times K}$ is the all-ones vector; λ_i is taken as an adjusting factor; $\varepsilon_{\mu,l,i,m}$ and $\varepsilon_{\sigma,l,i,m}$ are the learning rates to be determined to assure updating efficiency and stability of the learning process.

The increment of performance index $\bar{J}_{i,m,k}$ can be derived from (19) to be

$$\Delta \bar{J}_{i,m,k} = \int_a^b 2(\gamma_{i,m,k}(y) - \gamma_{i,O,k}(y))\Delta\gamma_{i,m,k}(y)dy \quad (24)$$

where $\Delta \bar{J}_{i,m,k} = \bar{J}_{i,m,k} - \bar{J}_{i,m-1,k}$, and $\Delta\gamma_{i,m,k}(y) = \sum_{l=1}^{n} w_{l,i,m,k}\Delta R_{l,i,m}$.

$$
\begin{aligned}
\Delta R_{l,i,m}(y) &= R_{l,i,m}(y) - R_{l,i,m-1}(y) \\
&\approx \frac{\partial R_{l,i,m-1}(y)}{\partial \mu_{l,i,m-1}}\Delta\mu_{l,i,m-1} + \frac{\partial R_{l,i,m-1}(y)}{\partial \sigma_{l,i,m-1}}\Delta\sigma_{l,i,m-1} \\
&= \frac{y - \mu_{l,i,m-1}}{\sigma_{l,i,m-1}^2}R_{l,i,m-1}(y)\Lambda_{\mu,l,i,m}\bar{\mathbb{E}}_{i,m-1} \\
&+ \frac{(y - \mu_{l,i,m-1})^2}{\sigma_{l,i,m-1}^3}R_{l,i,m-1}(y)\Lambda_{\sigma,l,i,m}\bar{\mathbb{E}}_{i,m-1}
\end{aligned}
\quad (25)
$$

Following Remark 2, the condition of $\Delta \bar{J}_{i,m,k} \leq 0$ is expanded as

$$
\begin{aligned}
&\int_a^b (\gamma_{i,m,k}(y) - \gamma_{i,O,k}(y)) \\
&\times \sum_{l=1}^{n} w_{l,i,m,k}\left(\frac{y - \mu_{l,i,m-1}}{\sigma_{l,i,m-1}^2}R_{l,i,m-1}(y)\Lambda_{\mu,l,i,m}\bar{\mathbb{E}}_{i,m-1}\right. \\
&\left. + \frac{(y - \mu_{l,i,m-1})^2}{\sigma_{l,i,m-1}^3}R_{l,i,m-1}(y)\Lambda_{\sigma,l,i,m}\bar{\mathbb{E}}_{i,m-1}\right)dy \leq 0
\end{aligned}
\quad (26)
$$

Moreover, if

$$\int_a^b (\gamma_{i,m,k}(y) - \gamma_{i,O,k}(y))\sum_{l=1}^{n}\Psi\Pi\Lambda_{\mu,l,i,m}dy \leq 0$$

and

$$\int_a^b (\gamma_{i,m,k}(y) - \gamma_{i,O,k}(y))\sum_{l=1}^{n}\Psi\Pi^2\sigma_{l,i,m-1}\Lambda_{\sigma,l,i,m}dy \leq 0$$
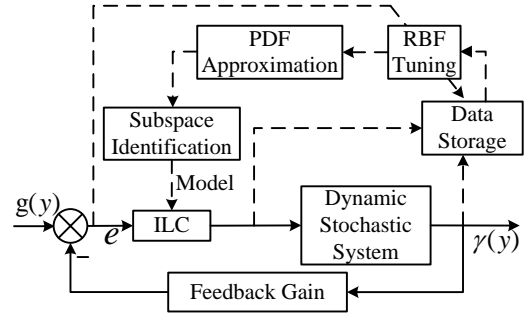

Fig. 1. Block diagram of the double closed-loop ILM and ILC System (solid lines: ILC, dashed lines: ILM)

where $\Pi = \frac{y - \mu_{l,i,m-1}}{\sigma_{l,i,m-1}^2}$ and $\Psi = w_{l,i,m,k}R_{l,i,m-1}(y)\bar{\mathbb{E}}_{i,m-1}$, then the convergence condition (26) will be satisfied.

The weights vector $V_{i,M,k} := [\omega_{1,i,M,k}, \omega_{2,i,M,k}, \cdots, \omega_{n-1,i,M,k}]$ can be calculated from (5) using the M-th tuned RBFNN, and a state-space model (6) can be established by the subspace identification as shown in Section II.B. This state-space model will be used for controller design to achieve the PDF tracking performance.

The updating of the RBFNN model parameters is actually an adaptive tuning process for modeling. If the whole process is divided into several modeling intervals (*BATCH*es), then the modeling accuracy in terms of PDF approximation is increased from *BATCH* to *BATCH*. The update of RBFNN parameters is also regarded as IL modeling (ILM).

Up to now the IL-based modeling and the IL-based control are taken as two separate matters. The ILC is implemented in the time domain. The ILM is undertaken in both the probability distribution domain and the time domain. For a SDC problem, the ultimate goal is to reduce the tracking errors in output PDF. For this purpose, a novel strategy that integrates ILM and ILC in one framewrok is proposed in the next section.

IV. IL-BASED DOUBLE CLOSED-LOOP MODELING AND CONTROL

A. Modeling and Control in Two Closed Loops

The idea of putting ILC and ILM in the same framework is to improve both the modeling accuracy and the control performance through iterative learning. This is particulary useful when the system output goes through large variations where the re-modeling is required to be adaptive. A simple way to combine ILC and ILM is to implement both with the same iterative period, and update the model and the controller together in each iteration. This, however, will be computationally inconvenient since the tuning periods required for modeling and control could be quite different. In general, the modeling update should take a longer period than the controller IL tuning. In this work, the ILC design is taken as a fast process and the ILM as a slower process. As illustrated in Fig.1, the ILC is put in the inner tuning loop and the ILM is arranged in the outer loop. In the inner loop is a standard ILC design based on the developed state-space PDF model, and in the outer loop is the ILM for the update of the RBFNN PDF approximation which has a longer iterative period.
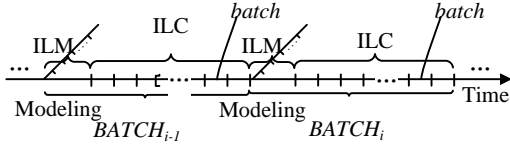
Fig. 2. Time period division of the iterative cycles in modeling and control

Remark 3: In this IL-based double closed-loop modeling and control framework, the targets of ILC and ILM are the same, both meant to drive the output PDF towards a target PDF. The two tracking errors, as defined in (12) for controller design and in (19) to (20) for model update, are used as the errors in ILC and ILM, respectively, that should be decreased through iterations.

Following the structure of the double closed-loop IL design, the whole time horizon is firstly divided into several intervals for ILM with a large period, denoted as *BATCH*es. In each *BATCH*, the time horizon is further divided into a modeling period plus several ILC intervals of the same time length, the latter are called *batch*es. The first modeling period is used to update the RBF parameters, which is to re-tune the center and width parameters of RBFs so that the average PDF approximation error in this *BATCH* is smaller than the previous *BATCH*. This update can be implemented several times (M times) to reach the modeling accuracy. With the updated RBFs, new weights for PDF approximation are calculated, the subspace identification method is used to re-establish the space-state model using the new weights and the collected control inputs. Note that in the modeling period, the control law stays the same as that from the final *batch* in the previous *BATCH*. After the modeling period, in the ILC *batch*es, the fixed (updated) model is used for ILC to drive the output PDF towards the desired PDF. In this way, the IL-based modeling and control are integrated into one framework where ILM is in the outer loop and ILC in the inner loop. Fig. 2 illustrates the division of modeling and control cycles in the time horizon.

The algorithms for ILC and ILM are similar to those presented in Section III. In the double closed-loop structure, rewrite the state-space model with subscripts i and j added to indicate the i-th *BATCH* and the j-th *batch*:

$$\begin{cases} x_{i,j,k+1} = A_i x_{i,j,k} + B_i u_{i,j,k} \\ V_{i,j,k} = C_i x_{i,j,k} + D_i u_{i,j,k} \\ \gamma(y, u_{i,j,k}) = \Lambda_i(y) V_{i,j,k} + L_i(y) \end{cases} \quad (27)$$

The update of the RBFNN parameters can be referred to Section III.B. The ILC design is similar to Section III.A in general, with only the tiny difference of introducing the i-th *BATCH* in the algorithm description as briefed in the following.

From the state-space representation in (27), the weights vector can be calculated as follows,

$$V_{i,j,k} = C_i A_i^k x_{i,j,0} + \sum_{s=0}^{k-1} C_i A_i^{k-1-s} B_i u_{i,j,s} + D_i u_{i,j,k} \quad (28)$$

Denoting

$$h_{i,0} = D_i$$
$$h_{i,k} = C_i A_i^{k-1} B_i$$

$$\mathbb{U}_{i,j} = [u_{i,j,0}^{\mathrm{T}}, u_{i,j,1}^{\mathrm{T}}, \cdots, u_{i,j,k}^{\mathrm{T}}, \cdots, u_{i,j,L}^{\mathrm{T}}]^{\mathrm{T}}$$
$$\mathbb{V}_{i,j} = [V_{i,j,0}^{\mathrm{T}}, V_{i,j,1}^{\mathrm{T}}, \cdots, V_{i,j,k}^{\mathrm{T}}, \cdots, V_{i,j,L}^{\mathrm{T}}]^{\mathrm{T}}$$
$$\mathbb{V}_{i,j,0} = [C_i^{\mathrm{T}}, (C_i A_i)^{\mathrm{T}}, \cdots, (C_i A_i^k)^{\mathrm{T}}, \cdots, (C_i A_i^L)^{\mathrm{T}}]^{\mathrm{T}} x_{i,j,0}$$

$$G_i = \begin{bmatrix} h_{i,0} & & & & \\ h_{i,1} & h_{i,0} & & \mathbf{0} & \\ h_{i,2} & h_{i,1} & h_{i,0} & & \\ \vdots & \vdots & \vdots & \ddots & \\ h_{i,L} & h_{i,L-1} & \cdots & h_{i,1} & h_{i,0} \end{bmatrix}$$

then (28) can be written in a compact form as follows:

$$\mathbb{V}_{i,j} = \mathbb{V}_{i,j,0} + G_i \mathbb{U}_{i,j} \quad (29)$$

where the initial values in $V_{i,j,0}$ are kept the same in each control iteration. Similarly, a quadratic performance index is used to evaluate the control performance.

$$J_{i,j+1} = \|\mathbb{E}_{i,j+1}\|_Q^2 + \|\mathbb{U}_{i,j+1} - \mathbb{U}_{i,j}\|_R^2$$
$$= \mathbb{E}_{i,j+1}^{\mathrm{T}} Q \mathbb{E}_{i,j+1} + (\mathbb{U}_{i,j+1} - \mathbb{U}_{i,j}^{\mathrm{T}}) R(\mathbb{U}_{i,j+1} - \mathbb{U}_{i,j}) \quad (30)$$

where

$$\mathbb{E}_{i,j} = \left[V_{i,j,0}^{\mathrm{T}} - V_{i,g}^{\mathrm{T}}, V_{i,j,1}^{\mathrm{T}} - V_{i,g}^{\mathrm{T}}, \cdots, V_{i,j,L}^{\mathrm{T}} - V_{i,g}^{\mathrm{T}} \right]^{\mathrm{T}} \quad (31)$$

is the vector of PDF tracking errors (written in the form of weights) in the i-th *BATCH*. The ILC law for the double closed-loop structure is therefore

$$\mathbb{U}_{i,j+1} = \mathbb{U}_{i,j} - [G_i^{\mathrm{T}} Q G_i + R]^{-1} G_i^{\mathrm{T}} Q \mathbb{E}_{i,j} \quad (32)$$

Due to the parameter tuning by ILM, the RBFNNs are actually different in each *BATCH* leading to different desired weights vector in each *BATCH* although the target PDF is the same. The termination condition of this ILC can be set by a tracking error threshold, i.e., $\|\mathbb{E}_{i,j}\|_Q^2 - \|\mathbb{E}_{i,j+1}\|_Q^2 \leq \delta_0$ (where δ_0 is a small positive number), or a fixed iteration number for control update. In this work, a fixed iteration number is introduced, but the ILC will terminate when reaching the threshold condition (before reaching the largest iteration number) in each *BATCH*. The work flow of the double-loop IL structure is shown in Fig.3. The main steps of the proposed method is summarized as follows.

1) The saved data of the output PDFs and control inputs from the previous *BATCH* are used to tune the RBF parameters by IL mechanism (23) (The RBF parameters in the first *BATCH* are given).
2) Using the updated RBFNN, the weights vector for PDF apptoximation is re-calculated from the saved data of output PDFs, and the state-space model of the output SDC system is established using the subspace identification technique as described in Section II.B (Model parameters for the first *BATCH* are estimated directly by random input excitation). Following the modeling period in each *BATCH*, the established state-space model will be used for all the afterwards ILC *batch*es within the same *BATCH*.
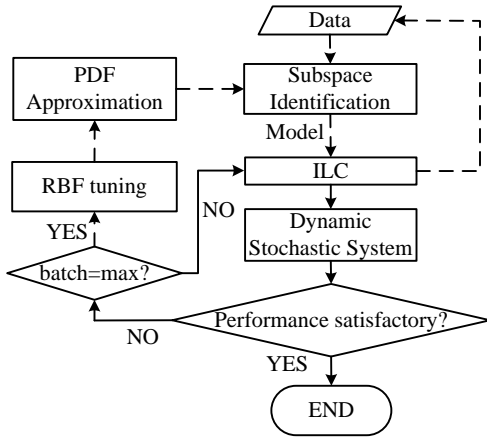
Fig. 3. Work flow of the double closed-loop ILM and ILC system

3) In the ILC *batch*es, the PDF controller is iteratively updated so as to achieve the closed-loop PDF tracking performance. All of the output PDFs and the control inputs are collected and saved to be used in the next *BATCH* to update the model.
4) Terminate the double closed-loop iteration if the PDF tracking performance is satisfactory or reaching the end of the *BATCH*, otherwise go back to 1) to move to the next *BATCH*.

Remark 4: Two factors are crucial to the data-based modeling in this work. One is that the data should contain adequate information for parameter identification. This may be solved to some extent by collecting sufficient data in each *BATCH*. The other is the RBFNN that provides the model structure for PDF approximation. If the RBF parameters are not properly chosen, the PDF approximation error could be large, then the parameter identification based on the poor weights calculation won't be able to faithfully describe the characteristics of the dynamic system. For this reason, the iterative update of RBF parameters is important in assuring the SDC control performance.

Remark 5: One role of the dynamic model is to support ILC design so as to improve the PDF tracking performance. The modeling error is not directly considered in the control performance evaluation. To save the computational efforts in modeling, there's no need to fine tune the model to reach the possible minimum modeling error, instead, we can either set up a reasonable time bound for the modeling period, or set up a threshold for the modeling error to terminate the modeling process in each *BATCH*.

Remark 6: The length of each individual ILC can be set following the time characteristics of the dynamic system together with the trial-and-error efforts. The interval of ILM is not predetermined. It is the result of the modeling period plus the number of ILC actually taken multiplied by the ILC interval. The interval of ILM is therefore different from *BATCH* to *BATCH*.

B. The Identifiability Issue

The double closed-loop structure for ILM and ILC includes a 'closed-loop' subspace identification process in the modeling

period. For most of the closed-loop systems, the noise in the output will affect the input signal through the feedback channel, therefore the input noise is related to the output noise. This could result in a non-identifiability problem for parameter estimation [26]. However, in this double closed-loop structure, both the inner loop and the outer loop are developed based on iterative learning algorithms, which has some special features to avoid or reduce the levels of non-identifiability problem.

(1) In each control *batch*, the PDF tracking errors are not instantly used to form the closed-loop control input. Instead, they are collected during the *batch*, only used to produce the control actions in the next *batch*. This makes a delay of one *batch* period between the output tracking error and the control input calculation, therefore largely reduces the noise correlations between the input and output signals. In another word, an 'open-loop' control is applied in each individual *batch* although the control is run in closed-loop in the whole *BATCH*.
(2) Each modeling *BATCH* contains a modeling period and a number of ILC *batch*es. In order to meet the requirements of ILC, in each *batch* initialization, the control inputs are set to the same initial values rather than taking the values from the end of the previous *batch*. The control input handled this way can be regarded as a man-made noise introduced into the system with a periodic step change, as illustrated in Fig.4.
(3) During the whole process horizon, the model parameters are periodically changed by the iterative learning update. At the beginning of $BATCH_i$, new model parameters (A_i, B_i, C_i, D_i) are obtained that provide the key information for ILC design in each *batch* afterwards. Therefore, the controller design is also a process with periodical 'interruptions'. See Fig.5 for an illustration of this strategy.
(4) Following the iterative learning update of the RBFNN, the target weights vector also goes through a change following the *BATCH* period even though the target PDF $\gamma_g(y)$ is kept unchanged. In the modeling period of the i-th *BATCH*, the target weights vector, $V_{i,g}$, is calculated using the updated RBFNN and applied to formulate the PDF tracking error in (31). The periodic update of $V_{i,g}$ is shown in Fig.6.
(5) There are always PDF modeling errors using RBFNN or any other nonlinear approximations. This error, together with the measurement noise, can be seen as the noise input introduced to the control system through the closed-loop feedback channel.

It can be seen from the above discussions that this double closed-loop IL structure has its set point (weights vector for the target PDF) going through periodic changes. The controller is nonlinear, time-varying, and iteratively updated. There's a delay in the feedback channel in terms of using the output PDF for closed-loop feedback control. The nonlinear modeling errors can be taken as continuous excitable noise input from the feedback channel, which is not correlated to the noise in the feedforward channel. Therefore it can be reasonably stated that the system is practically identifiable from the engineering
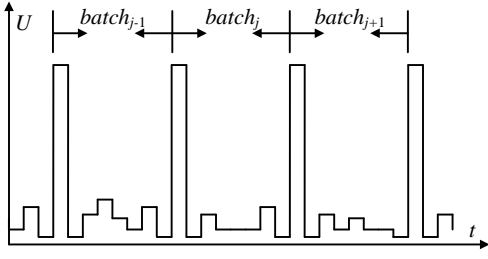
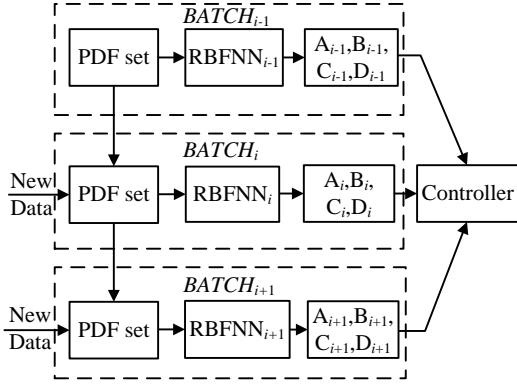Fig. 4. 'Periodic' interference of the control input signal in ILC cycles

Fig. 5. Periodic update of the state-space model parameters in ILM cycles

application point of view [27].

V. SIMULATION STUDY OF AN NUMERICAL EXAMPLE

In this section, a 4th-order SDC system is used in simulation study to examine the effectiveness of the proposed double closed-loop IL algorithm.

A. RBFNN-based PDF Modeling

In the output SDC systems, the output PDF $\gamma(y, u_k)$, as the variable to be controlled, should be measured or estimated by using instruments (for example, digital camera) or Bayesian estimation technique [30]. In this simulation, the inputs u_k and the output PDFs $\gamma(y, u_k)$ are assumed to be known. Consider the following model:

$$\begin{cases} x_{k+1} = A_0 x_k + B_0 u_k \\ V_{k,0} = C_0 x_k + D_0 u_k \\ \gamma_0(y, u_k) = \Lambda_0(y) V_{k,0} + L_0(y) \end{cases} \quad (33)$$

where the parameter matrices are given to be

$$A_0 = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ -1.0 & -5.0 & -3.0 \end{bmatrix}, \quad C_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 0.5 & 0.3 & 0.0 \\ 0.0 & 0.9 & 0.1 \\ -0.5 & -0.2 & 0.1 \end{bmatrix}, \quad D_0 = 0.$$

In this numerical example, the output PDFs of the SDC system are produced by a three-layer neural network with four RBFs defined by (2). $V_{k,0} = [\omega_{1,k,0}, \omega_{2,k,0}, \omega_{3,k,0}]^\top$, $y \in [-3, 3]$, $\Lambda_0(y) = [\lambda_{1,0}, \lambda_{2,0}, \lambda_{3,0}]$ and $L_0(y) = r_{4,0}^{-1} R_{4,0}(y)$ with
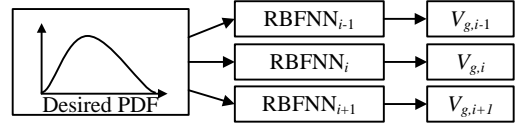
Fig. 6. Periodic change of the target weights vector under the same γ_g

$r_{l,0} = \int_{-3}^{3} R_{l,0}(y) dy$ and $\lambda_{l,0} = R_{l,0}(y) - R_{4,0}(y) r_{l,0}/r_{4,0}$ ($l = 1, 2, 3$). Assuming the RBF parameters to be $\mu_{1,0} = -1.5, \mu_{2,0} = -0.5, \mu_{3,0} = 0.5, \mu_{4,0} = 1.5$, and $\sigma_{1,0} = \sigma_{2,0} = \sigma_{3,0} = \sigma_{4,0} = 0.4$. The initial weights vector is set to be $V_{0,0} = [0.10, 0.50, 0.05]^\top$. The weights vector for the desired PDF is set to be $V_{g,0} = [0.2, 0.0, 0.6]^\top$. Model (33) will be used to produce the pseudo-measurement data for modeling.

In this simulation, the number of RBFs is also chosen to be 4, which is the same as the RBFNN model used to produce the simulation data. In the first *BATCH*, the RBF parameters are chosen as $\mu_{1,1} = -1.2, \mu_{2,1} = -0.2, \mu_{3,1} = 0.2, \mu_{4,1} = 1.2$, and $\sigma_{1,1} = 0.3, \sigma_{2,1} = 0.4, \sigma_{3,1} = 0.3, \sigma_{4,1} = 0.4$. The PDF model structure is given in (27). The order of the state-space model is chosen to be 4, and the subspace identification algorithm used in this simulation is *N4SID*. In parameter estimation of the first *BATCH*, a random excitation signal (pseudo-random binary sequence (PRBS)) is used as the input signal. The identified parameter matrices (A_1, B_1, C_1 and D_1) are as follows:

$$A_1 = \begin{bmatrix} 0.9011 & -0.1213 & 0.0891 & 0.0265 \\ -0.4217 & 0.1673 & 0.3658 & 0.0553 \\ 0.1391 & -0.2629 & 0.8538 & -0.1104 \\ -0.0679 & -0.7310 & -0.3301 & -0.1200 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0.0057 & 0.0352 & 0.0036 \\ -0.0102 & 0.1437 & 0.0172 \\ -0.0554 & -0.0584 & -0.0021 \\ -0.0796 & 0.0451 & 0.0305 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 9.2949 & -0.7030 & -7.1057 & 0.7292 \\ -0.5191 & 7.0259 & 0.5729 & 0.2517 \\ -1.9479 & -7.7595 & 1.6298 & 0.6037 \end{bmatrix}$$

$$D_1 = 0$$

Remark 7: It's important to make a proper calculation in the first *BATCH*, otherwise there might be numerical problems growing in the RBFNN update in the subsequent *BATCH*es. The modeling quality in the first *BATCH* is mainly assured through the large amount of random excitation data in input signal. In the subsequent *BATCH*es, the input and output data used for parameter estimation are collected from the previous *BATCH*.

Remark 8: The order of the state-space model is recommended to be $n - 1$ or n when the number of RBFs is n and $n \geq d$. In this simulation, the model order can also be 3.

B. Simulation of ILM

The simulation is divided into several ILM *BATCH*es ($i = 1, 2 \cdots$), and each *BATCH* contains one modeling period plus 8 ILC *batch*es ($j = 1, \cdots, 8$). The control horizon of each *batch* has 50 sampling time points ($L = 50$). The total number

TABLE I
IL UPDATE OF RBF PARAMETERS USING A 4TH-ORDER MODEL

i	$(\mu_{1,i},\sigma_{1,i})$	$(\mu_{2,i},\sigma_{2,i})$	$(\mu_{3,i},\sigma_{3,i})$	$(\mu_{4,i},\sigma_{4,i})$
1	(-1.200,0.300)	(-0.200,0.400)	(0.200,0.400)	(1.200,0.300)
2	(-1.204,0.303)	(-0.199,0.412)	(0.213,0.412)	(1.324,0.381)
3	(-1.216,0.310)	(-0.196,0.431)	(0.239,0.439)	(1.385,0.423)

of RBFNN update in the modeling period is set to be 20 ($M = 20$), each modeling period contains $400 \times i$ samples ($K = 400 \times i$). There are 120 samples uniformly collected in the definition domain of $[-3, 3]$ for the output PDF. The RBF parameters are updated by the IL mechanism (23), where the adjusting factors λ_i are calculated as

$$
\lambda_i = \begin{cases} 10^{-ceil\left(\log_{10}\left(\sum_k \bar{J}_{i,k}\right)-2\right)}, & \log_{10}\left(\sum_k \bar{J}_{i,k}\right) \geq 1 \\[2ex] 10^{-3}, & -3 < \log_{10}\left(\sum_k \bar{J}_{i,k}\right) < 1 \\[2ex] 0, & \log_{10}\left(\sum_k \bar{J}_{i,k}\right) \leq -3 \end{cases}
$$

in which $ceil(X)$ rounds the elements of X to the nearest integers towards positive infinity. The learning rate values ε_μ and ε_σ of the i-th *BATCH* are chosen as follows,

$$
\begin{bmatrix} \varepsilon_{\mu,1,i} \\ \varepsilon_{\mu,2,i} \\ \varepsilon_{\mu,3,i} \\ \varepsilon_{\mu,4,i} \end{bmatrix} = \begin{bmatrix} -1.5 & 0 & 0 & 0 \\ 0 & -1.8 & 0 & 0 \\ 0 & 0 & -1.9 & 0 \\ 0 & 0 & 0 & -2.0 \end{bmatrix} \begin{bmatrix} \mu_{1,i-1} \\ \mu_{2,i-1} \\ \mu_{3,i-1} \\ \mu_{4,i-1} \end{bmatrix}
$$

$$
\begin{bmatrix} \varepsilon_{\sigma,1,i} \\ \varepsilon_{\sigma,2,i} \\ \varepsilon_{\sigma,3,i} \\ \varepsilon_{\sigma,4,i} \end{bmatrix} = \begin{bmatrix} -4.2 & 0 & 0 & 0 \\ 0 & -4.0 & 0 & 0 \\ 0 & 0 & -3.0 & 0 \\ 0 & 0 & 0 & -4.4 \end{bmatrix} \begin{bmatrix} \sigma^2_{1,i-1} \\ \sigma^2_{2,i-1} \\ \sigma^2_{3,i-1} \\ \sigma^2_{4,i-1} \end{bmatrix}
$$

The PDF tracking performance meets the given threshold requirement after 3 *BATCH*es. The RBFNN parameter updating results of the 20-th tuning in each *BATCH* are listed in Table I. The change of the 4 RBFs after 3 iterations are illustrated in Fig.7, in which the solid lines represent the RBFs in the 1st *BATCH* and the dash-dot line indicate the RBFs in the 3rd *BATCH*. Figure 8 shows the output PDF approximation errors in the second and third *BATCH*es. The improvement in PDF approximation accuracy is clearly seen after 3 iterations.

In the third *BATCH*, the state-space model is estimated to have the following matrices.

$$
A_3 = \begin{bmatrix} 0.5533 & 0.6243 & -0.2236 & 0.0342 \\ 0.2112 & 0.2848 & 0.4885 & -0.0334 \\ 0.0295 & -0.1089 & -0.0209 & -0.2085 \\ 0.0010 & -0.0811 & -0.0337 & 0.9772 \end{bmatrix}
$$

$$
B_3 = \begin{bmatrix} 0.5074 & -0.1384 & -0.0671 \\ 0.0870 & 1.2669 & 0.1729 \\ -0.1208 & -0.1976 & -0.0005 \\ 0.0497 & 0.1937 & 0.0247 \end{bmatrix}
$$

$$
C_3 = \begin{bmatrix} 0.5463 & 0.6447 & -0.1432 & 0.1082 \\ -1.3025 & 0.5536 & -3.4939 & 0.6568 \\ 1.2898 & -0.5326 & 4.6296 & -0.8293 \end{bmatrix}
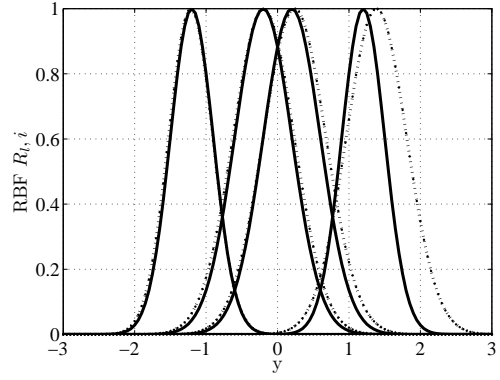$$

$$
D_3 = 0
$$

Fig. 7. The initial and the final RBFs (solid lines: $i = 1$, dotted lines: $i = 3$)
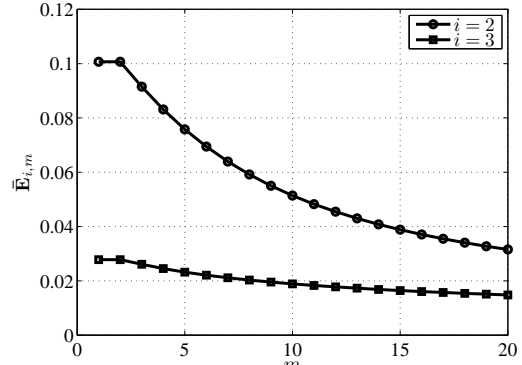

Fig. 8. The RBFNN approximation errors of output PDF during the modeling period ($q = 4$)

TABLE II
IL UPDATE OF RBF PARAMETERS USING A 3RD-ORDER MODEL

i	$(\mu_{1,i},\sigma_{1,i})$	$(\mu_{2,i},\sigma_{2,i})$	$(\mu_{3,i},\sigma_{3,i})$	$(\mu_{4,i},\sigma_{4,i})$
1	(-1.200,0.300)	(-0.200,0.400)	(0.200,0.400)	(1.200,0.300)
2	(-1.205,0.303)	(-0.202,0.417)	(0.211,0.412)	(1.326,0.382)
3	(-1.218,0.310)	(-0.203,0.438)	(0.232,0.436)	(1.389,0.426)
4	(-1.240,0.322)	(-0.207,0.465)	(0.252,0.461)	(1.423,0.443)
5	(-1.270,0.341)	(-0.212,0.490)	(0.269,0.480)	(1.448,0.446)

It can be found that (A_3, B_3, C_3) are quite different from (A_1, B_1, C_1).

The state-space model can also be described by other orders. When using a 3rd-order description in simulation, the PDF tracking performance meets the threshold requirement after 5 *BATCH*es. The parameter updating results after the 20-th tuning in all *BATCH*es are listed in Table II and the output PDF approximation errors in each *BATCH* is shown in Fig.9. It can be seen that good modeling results are achieved for both 3rd- and 4th-order state-space models.

C. Simulation of ILC

In the inner loop, there are 8 ILC *batch*es in each *BATCH*. The learning rate is chosen according to (32), where $Q = 2I_1$, $R = 0.001I_2$, I_1, I_2 are unit matrices of proper dimensions. The termination condition of ILC is given as $\sum_{k=1}^{50} \int_a^b (\gamma(y, u_{i,j,k}) - \gamma_g(y))^2 dy < 0.01$. A performance
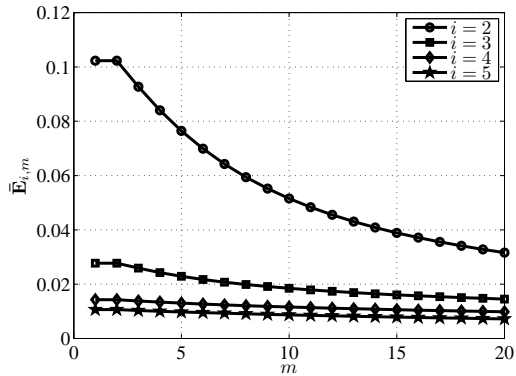
Fig. 9. The RBFNN approximation errors of output PDF during the modeling period in all BATCHes ($q = 3$)

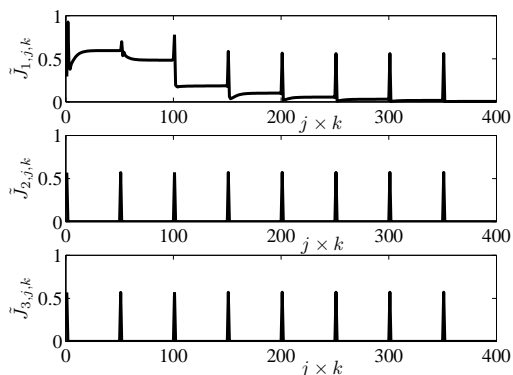Fig. 11. Dynamic evolution of output PDFs in the final *BATCH* (8 ILC batches included)

Fig. 10. The PDF tracking errors of all ILC batches in 3 ILM iterations (where $\tilde{J}_{i,j,k} := \int_a^b (\gamma_0(y, u_{i,j,k}) - \gamma_g(y))^2 dy$)
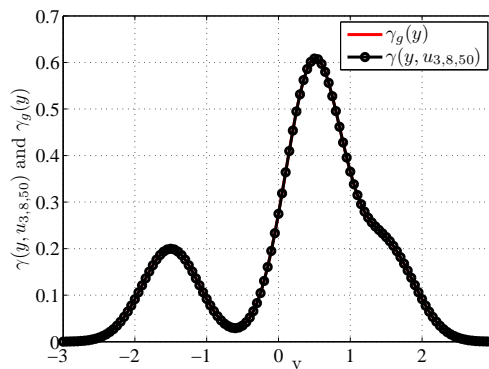
Fig. 12. The desired PDF and the final PDF at the end of the completed IL process

measure in the inner ILC loop is illustrated in Fig.10. Indeed, Fig.10 shows that the optimal ILC results in an efficient output PDF tracking and ensures the stability of the inner ILC loop. It can be seen that the performance of the first *batch* ($j = 1, k = 1, 2, \cdots, 50$) inside the first *BATCH* ($i = 1$) as shown in Fig.10 is not good before the ILC action. The evolution of the output PDFs in the third *BATCH* is shown in Fig.11. Also, the desired PDF and the final PDF in the 8-th *batch* of the third *BATCH* are illustrated in Fig.12, which shows a perfect tracking performance. As discussed in Section III.A, in this IL-based double closed-loop structure, the control cost function in (11) should always decrease monotonically. Fig.13 demonstrates the total ILC performance index of each *batch* in 3 *BATCH*es (the difference between the second and third *BATCH* is visually indistinguishable), which shows the convergence of the double closed-loop IL algorithm.

VI. APPLICATION STUDY: TEMPERATURE FIELD DISTRIBUTION CONTROL ON A VISUAL OPERATION PLATFORM OF A COAL COMBUSTION PROCESS

The idea of SDC is of particular interest to many industrial processes with controlled variables having a spatial distribution characteristics such as food processing, paper-making, polymerization, and combustion processes. However, due to the difficulties in online PDF measurement and the
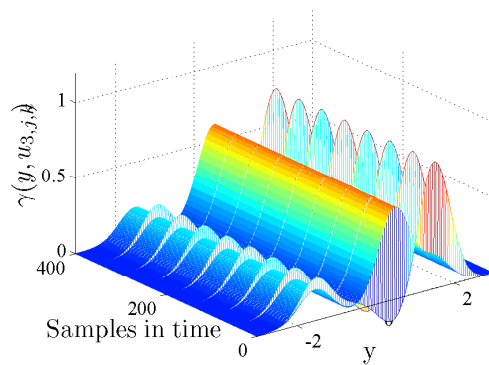
computational complexity of SDC modeling and control, the majority of work in this subject is still on the theoretical aspect and the practical application results are extremely rare. One feasible approach towards industrial applications is to use simulation technology to construct a visual platform where the simulation operation is very close to the actual system in operation. Based on this pseudo operation system, the SDC modeling and control strategy can be implemented to the full
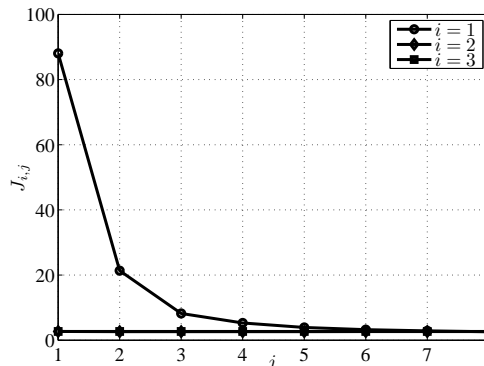

Fig. 13. The control performance index in all *BATCH*es (8 ILC iterations in each *BATCH*)
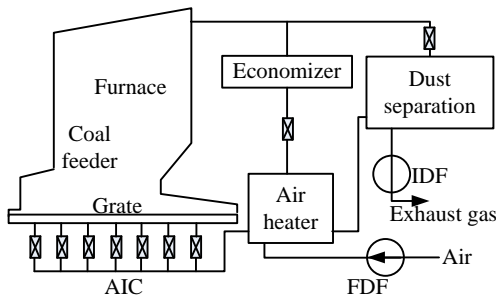
Fig. 14. The boiler coal combustion system, IDF: Induced draft fan; FDF: Forced draft fan; and AIC: Air inlet chambers

operation range. In this section, we first construct a visual simulation platform for the system of chain coal-layer-fired boilers as shown in Fig.14, in which the furnace temperature field distribution is the system output to be controlled. We then apply the double-loop IL modeling and control to this system to investigate how the system performance can be improved through the iterative learning process.

A. The Visual Simulation Platform

A visual simulation platform used for the purpose of operational training, education or real-time analysis generally has two parts. One is the numerical simulation of the real system, which often includes a dynamic model with regards to key parameters of the system. The other is the visualization of the operational process, which is established through a real-time simulation interface based on the model calculation. There are various visualization tools available such as *OpenGL* and *3Dmax*, also configurable softwares available such as *WinCC*, *InTouch* and *KingView*, etc. These industrial control software are particularly designed for automation control engineers, and are widely applied in the field of process control [36]–[39].

The configuration software is a specialized software for data acquisition and process control. It can be used as a software development environment to support a variety of industrial equipments and common communication protocols, and also enables users to create their own human machine interface (HMI). In this work, a configurable simulation system is used to develop the visual simulation platform for the chain grate boiler system with coal combustion. It provides an easy configuration of the main system functions, a close simulation to the real operating system and a simple evaluation of the real-time operating data. This simulation platform can be conveniently extended to a variety of simulation platforms, and used as simulators for education and training of control operations of similar boiler combustion processes. As shown in Fig.15, WinCC, MATLAB and the OPC communications are used to establish the simulation interface, programme the model and control algorithms, and exchange data between the interface and model.

The boiler coal combustion system is shown in Fig. 14. The configurations of the boiler system and the furnace are shown in Fig.16 and Fig.17, respectively. In this system, the main controlled variable is the furnace field temperature distribution, which is to a large extent, closely related to the flame shape
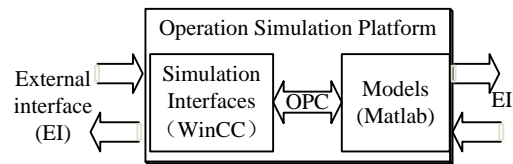

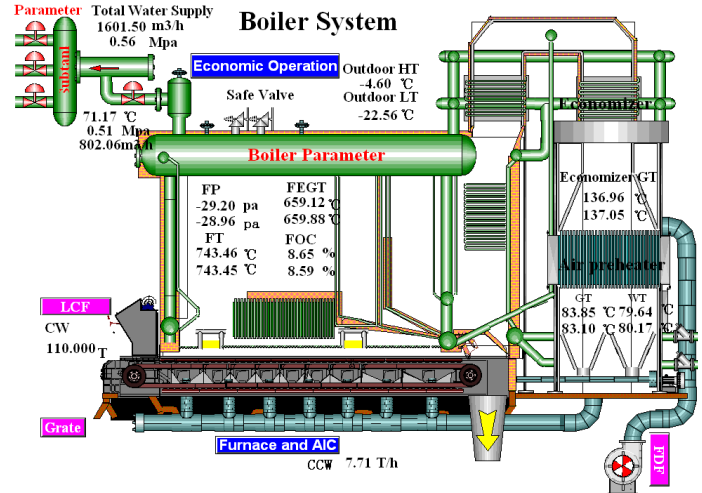Fig. 15. The overall platform structure for operation simulation

Fig. 16. Operational interface of the boiler system

and color inside a furnace that can be physically measured during the combustion process. The real-time distribution produced from flame images are therefore taken as the output PDF to be controlled, and the feedback signals are provided by a charge-coupled device (CCD). In an ideal furnace, the combustion air flow should be matched to the fuel flow to assure complete combustion. In the real world, combustion does not proceed in a perfect manner, therefore more air (oxygen) is provided than it is theoretically estimated. The temperature field distribution in the chain grate boilers is mainly controlled by the coal feeder, the air inlet and the grating. The openness of air inlet chambers determines the total inlet air flow and the air-to-coal ratio, which is regarded as a key operating factor that affects the combustion quality. We select the inlet air chamber openness as the control input in this system.

B. Coal and Flame Model

Model development of the coal combustion process is the main challenge in establishing this visual simulation platform. The chain boiler combustion process can be divided into the bed combustion and the furnace gaseous combustion, among which the former is the key process. A full mechanism model of a bed combustion process is in general rather complicated and the detailed modeling can be found from literature [31]–[35]. In this work, a simplified combustion model is developed to support the implementation of the double closed-loop IL algorithm. The data flow diagram of modeling and control is shown in Fig.18.

The so-called "zoned model" or "cell model" [40] is employed to describe the combustion process. Here the entire
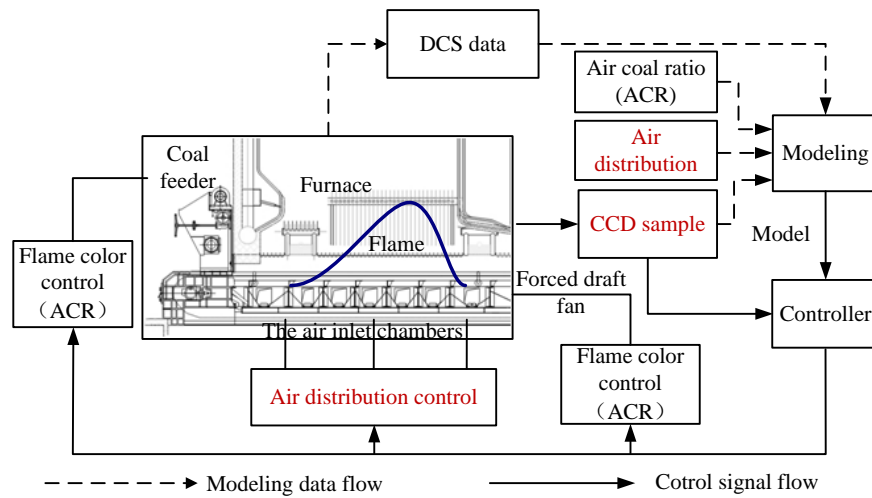
Fig. 18. Modeling and control data flow of the combustion process. ACR: air-coal ratio; CCD: charge-coupled devices.
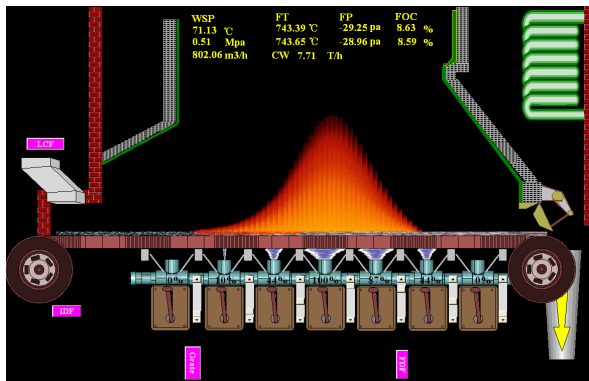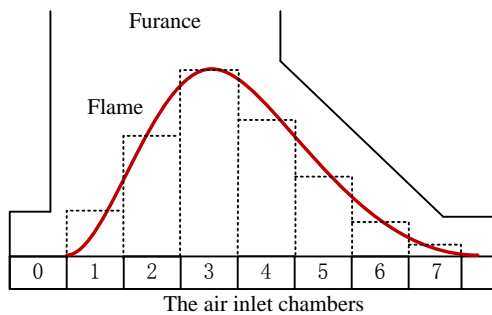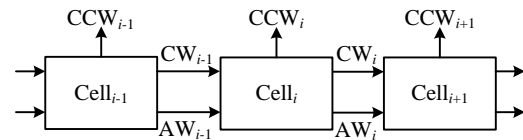

Fig. 17. Operational interface of the furnace

Fig. 20. Chamber mass conservation: CW - coal weight, AW - ash (CCR) weight, CCW - consumed coal weight

The following assumptions are made to establish the simplified zoned model.

1) The flows of gas and solids inside the furnace are upwards only. There is no return flow downwards.
2) The porosity rate in each cell is uniformly distributed. No radial variations in porosity are considered.
3) The particle size distribution and the quality of the coal are uniform through all the burning process.
4) The inner convective heat transfer is relatively small that can be ignored in calculation.
5) The rate of gas and solid flow change is much higher than the rate of heat transfer or the flow rate of each gas component. The dynamic change of the gas and solid flows is not considered. A static flow model is used instead.
6) In each cell, unburned coal and coal combustion residues (CCRs) are the only two materials in solid phase. The coal combusts into CCR by a fixed ratio.

Fig. 19. Zone division of the furnace

furnace is divided into a series of separate but inter-connected zones, each of the sub-zones is also called a cell. The laws of mass balance and energy balance apply in each cell. Following an industrial process of a "DZL58-1. 6/150/90-AII" boiler, the division modal of the air inlet zones is illustrated in Fig.19, in which each cell corresponds to an air inlet chamber. When coal is evenly fed to the furnace, the three-dimensional flame shape can be reduced to two-dimensions, thus the height of a rectangle in Fig.19 is taken as the height of flame in this zone.

The quantity of coal on the grate consists of two parts. One is the original unburned coal, m_c, the other is the CCRs, m_z, which are the materials remained after burning. Assume for every kilogram of coal, there are A_z amount of CCR produced after the burning. According to the mass balance law, the ordinary differential equation model for m_c and m_z in the

i-th cell can be written as follows.

$$\frac{dm_{c,i}}{dt} = P_{io}(m_{c,i-1} - m_{c,i}) - P_{cr,i}m_{c,i} - P_{lo}m_{c,i}$$

$$\frac{dm_{z,i}}{dt} = P_{io}(m_{z,i-1} - m_{z,i}) - P_{cr,i}m_{c,i}A_z - P_{lo}m_{z,i}$$

$$(34)$$

where $m_{c,i-1}$ and $m_{z,i-1}$ are the quantities of unburned coal and CCR transferred, over a unit time, from the $(i-1)$-th cell, $P_{cr,i} \in [0,1]$ is the coal combustion rate in the current cell; P_{io} is a function of grate frequency and P_{lo} is the rate of leakage coal. In these parameters , $P_{cr,i}$ can be described with a parabolic function about air coal ratio (ACR) to give

$$P_{cr,i} = -K_{d2}\left(\frac{P_g K_{b,i}}{m_{c,i}}\right)^2 + K_{d1}\left(\frac{P_g K_{b,i}}{m_{c,i}}\right) + K_{d0} \quad (35)$$

where P_g is the frequency of forced draft fan, $K_{b,i}$ is the openness of the i-th air inlet chamber which directly indicates the inlet air flow in the furnace. K_{d0}, K_{d1}, K_{d2} are constants.

In the two–dimensions model, the flame distribution can be described by three parameters, i.e., height, width and color. The color of flame is controlled by the ACR. The height can be approximated by a proportion of the consumed coal weight, $m_{c,i}$, and the coal combustion rate, $P_{cr,i}$. The width can be taken as a two-value function of $m_{c,i}$.

$$H_{f,i} = K_f \cdot m_{c,i} \cdot P_{cr,i}$$

$$D_{f,i} = \begin{cases} 0, & m_{c,i} = 0 \\ 1, & m_{c,i} > 0 \end{cases}$$

where $H_{f,i}$ and $D_{f,i}$ are the height and the width of flame and $K_f > 0$ is a constant.

C. Output SDC on the Visual Simulation Platform

The purpose of control in this system is to drive the PDF of flame distribution to follow the desired PDF. The target PDF for flame distribution is set up in the operation panel for 'desired PDF' through HMI. Note that unlike the numerical example in Section V, the desired/target PDF in the flame system is not produced by a known RBFNN, instead, it is manually set up through HMI (See Fig.21, here only the 2nd to the 6th bars are used to adjust the desired PDF). All of the simulation parameters including those used for process modeling and those used for PDF modeling and controller design such as the number of RBFs and their parameters, sampling numbers, settings of ILM *BATCH* and ILC *batch*, etc., can be easily set up in a parameter adjustment panel through HMI (See Fig.22 as an example).

In order to build the dynamic model between the openness of the air inlet chambers (control input) and the flame distribution (system output), a PRBS signal is introduced to change the opening of the 2nd to 6th air inlet chambers (the 1st and the 7th air inlet chambers are shut down in normal operating conditions). A total number of 255 samples are collected for the initial modeling in the first *BATCH*. The other parameters of the boiler are given as follows, $m_{c,0} = 7.71T/th$, $m_{z,0} = 0T/h$, $K_f = 1$, $K_{b,1:7}(0) = [0,0,0,0,0,0,0]$ $K_{b,1:7}(\infty) = [0,0.1,0.44,1,0.87,0.44,0]$,
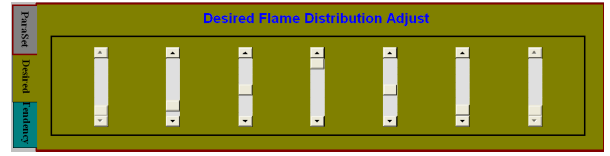

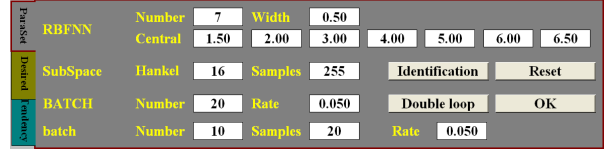Fig. 21.   The HMI panel for setting up of the desired flame distribution

Fig. 22.   The parameters setting HMI

$K_{d0} = 0.01$, $K_{d1} = 1.6/2.85$, $K_{d2} = 1$, $P_{io} = [15/19, 15/19, 18/19, 20/19, 18/19, 15/19, 15/19]$ and $P_{lo} = 0$.

Seven RBFs are used in modeling the flame PDF. They all have the same width, i.e., $\sigma_1 = \cdots = \sigma_7 = 0.5$. The maximum number of ILC *batch*es is set to be 10 and each *batch* has 20 samples in time horizon. The modeling period for RBFNN adjustment is also set to be 20. In the subspace identification, the numbers of rows and columns for Hankel matrices are $s = 16$ and $N = \text{length}(\mathbb{U}) - 14$, where $\text{length}(\mathbb{U})$ is the length of input vector $\mathbb{U}$. At the beginning of the IL process, the centers of the 7 RBFs are set to be $\mu_1 = 1.5, \mu_2 = 2, \mu_3 = 3.0, \mu_4 = 4.0, \mu_5 = 5.0, \mu_6 = 6.0, \mu_7 = 6.5$ (see Fig.22 for detail). Some simulation results are shown in Fig. 23 to Fig. 26.

Fig. 23 is a snapshot of the operational screen during the modeling period in the first *BATCH*, which presents the desired flame distribution, the real-time flame distribution and the PDF tracking errors. The simulation result in Fig. 24 shows that after the 9-th *batch* in the 4-th *BATCH*, the output PDF reaches the target PDF within the error threshold. A good tracking performance is achieved after 4 *BATCH*es. Note that there is a 'missing-data' period in Fig. 23 and 24 due to the operational system switch for discharging CCR from the furnace. For the same system, if we reduce the control time horizon within each *BATCH* to an inadequate level, for example, set the maximum number of ILC *batch*es to be 5, the system fails to achieve the PDF tracking performance even after going through several ILM *BATCH*es. See Fig.25 and 26 for the results after 3rd and 10-th *BATCH*es, respectively. This suggests that in this double closed-loop structure, the control horizon in each *BATCH* needs to be adequately long to fulfill the control purpose using the current modeling information.

VII. CONCLUSION

Compared with conventional stochastic control algorithms, an extra dimension for processing of probability space information is required in output SDC. In such a system, the output PDF to be controlled can be approximated by a RBFNN model. The selection of a suitable RBFNN model will directly affect the modeling accuracy of the dynamic PDF model, and indirectly affect the controller design. In this work, the IL
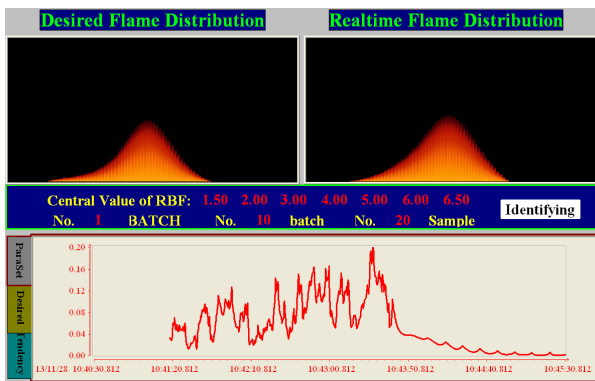
Fig. 23. Real-time operation screen 1: Top left - desired flame distribution; top right - 'measured' flame distribution; bottom - PDF tracking error signal during the parameter identification process in the 1st *BATCH*.
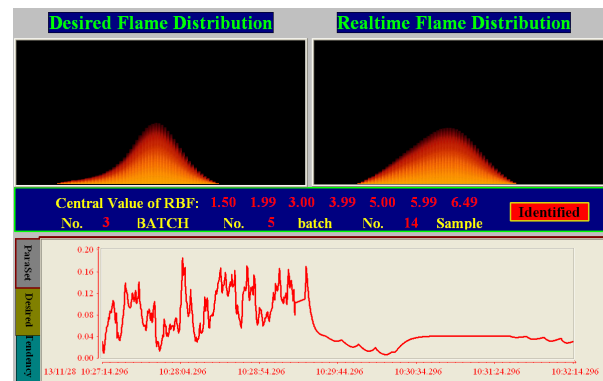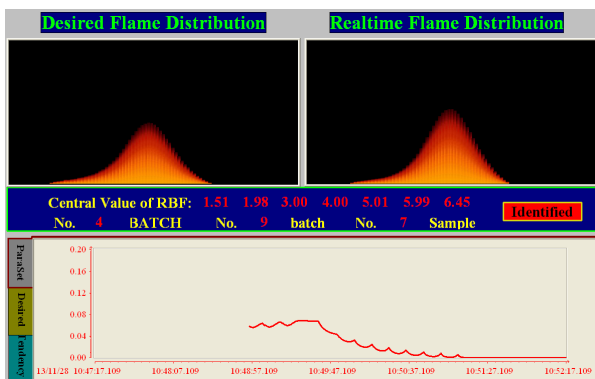

Fig. 24. Real-time operation screen 2: the bottom figure shows the PDF tracking error signal during the 9-th *batch* in the 4-th *BATCH*

Fig. 25. Real-time operation screen 4: the bottom figure shows the PDF tracking error signal during the 5-th *batch* in the 3rd *BATCH* when the maximum number of ILC iterations in each *BATCH* is set to be 5
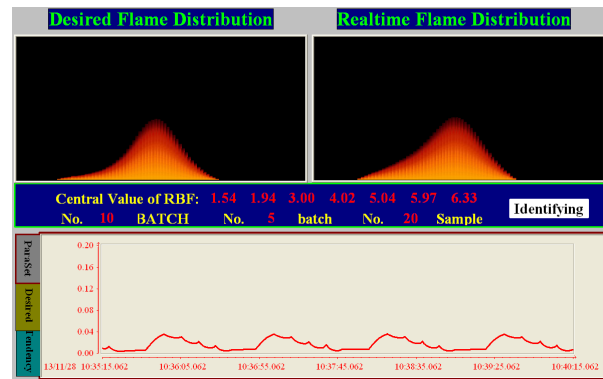

Fig. 26. Real-time operation screen 5: the bottom figure shows the PDF tracking error signal during the 5-th *batch* in the 10-th *BATCH* when the maximum number of ILC iterations in each *BATCH* is set to be 5

update of both the RBFNN parameters and the controller are integrated in the same framework that has a double closed-loop structure. The outer loop is an iterative learning modeling loop for updating the parameters of the RBFNN in the probability space. The inner loop is an ILC loop for updating controller in the time-domain. With this double closed-loop structure, the system model can be improved through the iterative learning update of the RBFNN parameters and the re-identification of the state-space model, and the output PDFs are controlled towards a target distribution through both ILC and ILM. The practical identifiability of this double closed-loop structure is discussed from the control engineering point of view to assure the feasibility of the closed-loop IL modeling.

The output PDF control problem should be considered for a general nonlinear stochastic system. It is difficult to achieve the control target by using a linear state-space modeling as presented in this paper [18], [21]. However, the double closed-loop structure can help to overcome this disadvantage, to some extent, by iteratively updating the linear model to increase the modeling accuracy. Simulation examples demonstrate that the double closed-loop IL modeling and control structure can effectively achieve the PDF tracking performance. A visual simulation platform of an industrial coal combustion process with chain boilers is developed to evaluate the modeling and control of the furnace temperature field distribution in operational environments. It can be seen from this pseudo-industrial example that the proposed double closed-loop IL algorithm can be implemented to tackle real-time SDC problems using a standard computational platform.

REFERENCES

[1] K.J. Astrom, *Introduction to Stochastic Control Theory,* New York Academic, 1970.
[2] B.D.O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods,* Englewood Cliffs, NJ: Prentice-Hall, 1989.
[3] H.B. Ji and H.S. Xi, "Adaptive output-feedback policy for nonlinear stochastic systems," *IEEE Trans. on Automatic Control*, vol. 51, pp. 355–360, 2006.
[4] H.N. Wu and K.Y. Cai, "Model-independent robust stabilization for uncertain Markovian jump nonlinear systems via fuzzy control," *IEEE Trans Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, pp. 509–519, 2005.
[5] D.W. Stroock, *An Introduction to Markov Processes,* World Publishing Company, 2009.
[6] J.C. Pinto, "A matrix representation of polymer chain size distributions, linear polymerization mechanisms at steady-state conditions," *Macromolecular Theory and Simulations*, vol.10, pp. 79–99, 2001.
[7] B. Alhamad, J.A. Romagnoli, and V.G. Gomes, "On-line multi-variable predictive control of molar mass and particle size distributions in free-radical emulsion copolymerization," *Chemical Engineering Science*, vol. 60, pp. 6596–6606, 2005.
[8] F.J. Doyle, C.A. Hartison and T.J. Crowley, "Hybrid model-based approach to batch-to-batch control of particle size distribution in emulsion polymerizatio," *Computers and Chemical Engineering*, vol. 27, pp. 1153–1163, 2003.
[9] H. Wang, *Bounded Dynamic Stochastic Systems: Modeling and Control,* Springer Verlag Ltd., 2000.

[10] M. Karny, "Towards fully probabilistic control design," *Automatica*, vol. 32, pp. 1719–1722, 1996.

[11] M.G. Forbes and J.F. Forbes, "Control design for first-order processes: Shaping the probability density of the process state," *Journal of Process Control*, vol. 14, pp. 399–410, 2004.

[12] A.P. Wang and H. Wang, "Minimum entropy control using B-spline square root models," *IEEE Proc. Control Theory and Applications*, vol. 151, pp.422–428, 2004.

[13] H. Wang and J.H. Zhang, "Bounded stochastic distribution control for pseudo ARMAX systems," *IEEE Trans. on Automatic Control*, vol. 46, pp.486–490, 2001.

[14] P. Afshar, H. Yue and H. Wang, "Robust iterative learning control of output PDF in non-Gaussian stochastic systems using Youla parameterization," in *Proceedings of the 2007 American Control Conference*, New York, USA.

[15] H. Wang, "Robust control of output probability density functions for multi-variable stochastic systems with guaranteed stability," *IEEE Trans. on Automatic Control*, vol. 44, pp. 2103–2107, 1999.

[16] J.L. Zhou and H.Wang, "Optimal tracking control of output PDF: Mean square root B-spline model," *Control Theory and Application*, in Chinese, vol. 22, pp. 369–376, 2005.

[17] H.Y. Chen, and H. Wang, "PDF Control of stochastic parameter system using linear matrix inequalities," *ACTA Automatica Sinica*, vol. 33, pp.1216–1220, 2007.

[18] L. Guo and H. Wang, *Stochastic Distribution Control Systems Design: A Convex Optimization Approach,* Springer-Verlag Ltd., 2010.

[19] L. Guo and L.P. Yin, "Robust PDF control with guaranteed stability for non-linear stochastic systems under modeling errors," *IET Control Theory Applications*, 3, p.575–582, 2009.

[20] L. Guo, L.P. Yin, H. Wang and T.Y. Chai, "Entropy optimization filtering for fault isolation of nonlinear non-Gaussian stochastic systems," *IEEE Trans. on Automatic Control*, vol. 54, pp.804–810, 2009.

[21] Y. Yi, L. Guo and H. Wang, "Adaptive statistic tracking control based on two steps neural networks with time delays". *IEEE Trans Neural Network*, vol.20, pp.420–429, 2009.

[22] H. Wang and P. Afshar, "ILC-based fixed-structure controller design for output PDF shaping in stochastic systems using LMI techniques," *IEEE Trans. on Automatic Control*, vol. 54, pp. 760–773, 2009.

[23] A.P. Wang, P. Afshar and H. Wang, "Complex stochastic system modeling and control via iterative machine learning," *Neurocomputing*, vol. 71, pp. 2685–2692, 2008.

[24] P.V. Overchee and B.D. Moor, *Subspace Identification for Linear Systems: Theory - Implementation - Applications,* Dordrecht: Kluwer Acadmic Publishers, 1996.

[25] T. Katayama, *Subspace Methods for System Identification,* London : Springer-Verlag, 2005.

[26] L. Ljung, *System Identification: Theory for the user,* Upper Saddle River: Prentice-Hall, 1999.

[27] D.Y. Xiao, "Experimental investigation of closed loop on-line real time identification," *Journal of Tsinghua University*, in Chinese, vol. 23, pp. 1-12, 1983.

[28] J.H. Lee and K.S. Lee, "Iterative learning control applied to batch processes: An overview," *Control Engineering Practice*, vol. 15, pp. 1306–1318, 2007.

[29] Y.Q. Wang, F.R. Gao and F.J. Doyle, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, pp. 1589–1600, 2009.

[30] M. Hazewinkel (Ed.), *Encyclopedia of Mathematics,* vol. 1 (A-B), Kluwer, 1987.

[31] J.J. Ji, Y.H. Luo and L.Y. Hu, "Study on the mechanism of unsteady combustion related to volatile in a coal-fired traveling grate boiler," *Applied Thermal Engineering*, vol. 28, pp. 145–156, 2008.

[32] K. A. Soteris, "Artificial intelligence for the modeling and control of combustion processes: a review," *Progress in Energy and Combustion Science*, vol. 29, pp.515–566, 2003.

[33] B. Labibi, H.J. Marquez and T.W. Chen, "Decentralized robust PI controller design for an industrial boiler," *Journal of Process Control*, vol.19, pp.216–230, 2009.

[34] Y.T. Chu, C. Lou, Q. Cheng and H.C. Zhou, "Distributed parameter modeling and simulation for the evaporation system of a controlled circulation boiler based on 3-D combustion monitoring," *Applied Thermal Engineering*, vol. 28, pp. 164–177, 2008.

[35] D.L. Chen et al., "Quasi-constant temperature combustion for improving the overall performance of a coal-fired boiler," *Combustion and Flame*, vol. 134, pp. 81–92, 2003.

[36] X. Ji, Y.W. Dong, Y.Y. Wang and H.B. Zhao, "Design and implementation of configurable simulation system for embedded software based on MDA," In *Proceedings of the 2010 Summer Computer Simulation Conference*, pp. 190–194, 2010.

[37] Y.Y. Wang, X.S. Zhou, Y.W. Dong and S. Liu, "A configurable environment simulation tool for embedded software," In *ATC'11 Proceedings of the 8th international conference on Autonomic and trusted computing*, pp. 47–59, 2011.

[38] R.K. Boyd and J.H. Kent, "Three-dimensional furnace computer modeling," In *21st Symp. (Int'l.) on Combustion*, pp. 265–274, 1986.

[39] H.T. Liu and H. Berndt, "Interactive design and simulation platform for flight vehicle systems development," *Journal of Aerospace Computing Information and Communication*, vol. 3, pp. 550–561, 2006.

[40] H.S. Xu and J.R. Zhu, "The dynamic modeling and simulation of a chain grate boiler," *Journal of Engineering for Thermal Energy and Power*, in Chinese, vol. 20, pp. 527– 531, 2005.

[41] J.R. Zhu, "Zoned model for a chain grate," *Journal of Engineering for Thermal Energy and Power*, in Chinese, vol. 15, pp. 50–51, 2000.

APPENDIX
NOTATIONS

Subscripts:

1) i - index for modeling *BATCH*es in the outer loop; (also used to indicate cells in furnace modeling)

2) j - index for control *batch*es in the inner loop;

3) k - index for sampling time in each *batch* or sample in each *BATCH*;

4) l - index for radial basis functions; $(l = 1, \cdots, n)$

5) m - index for RBFNN tuning times in the modeling period $(m = 1, \cdots, M)$;

6) Ordering sequence when several subscripts are used for the same variable: l, i, j, k or l, i, m, k.

Scalars:

1) $\gamma(y, u(t))$ - output PDF;

2) P - probability;

3) R_l - radial basis function;

4) μ_l, σ_l - RBF parameters for center and width;

5) n - total number of RBFs;

6) e - PDF approximation error term (ignored in modeling);

7) ε - error between the 'ideal' PDF and the PDF approximated by RBFNN;

8) s, N - number of rows and columns of Hankel matrices;

9) L - total number of sampling points in each ILC cycle;

10) K - total number of samples in the modeling period;

11) M - total updating times for the parameters of the basis functions in the time horizon for modeling

12) $J_{i,j}$ - ILC performance index in the j-th batch, i-th BATCH;

13) $\bar{J}_{i,m,k}$-ILM performance index at time k in the i-th BATCH after the m-th RBFNN tuning;

14) $\epsilon_\mu, \epsilon_\sigma$ - leaning rates for μ and σ in ILM;

15) q - order of the state-space model;

16) $\bar{E}_{i,m}$ - measure of the overall PDF approximation errors after the m-th RBFNN update in the modeling period within the i-th *BATCH*.

Vectors and matrices

1) $V \in \mathbb{R}^{p \times 1}$ - RBFNN weights vector, $V = [\omega_1, \cdots, \omega_{n-1}]^{\mathrm{T}}, p = n - 1$;

2) $x \in \mathbb{R}^{q \times 1}$ -process state vector;

3) $u \in \mathbb{R}^{d \times 1}$ -control input vector;

4) $A \in \mathbb{R}^{q \times q}, B \in \mathbb{R}^{q \times d}, C \in \mathbb{R}^{p \times q}, D \in \mathbb{R}^{p \times d}$ - state-space model parameter matrices;

5) $\bar{C} \in \mathbb{R}^{p \times q}, \bar{D} \in R^{p \times d}$ -state-space parameter matrices for the special case of no feed-forward loop;

6) $\mathbb{E}_j \in \mathbb{R}^{(p(L+1)) \times 1}$- output tracking error in the j-th ILC *batch*;

7) $Q \in \mathbb{R}^{(p(L+1)) \times (p(L+1))}, R \in \mathbb{R}^{(d(L+1)) \times (d(L+1))}$ - weighting matrices for $\mathbb{E}$ and $\mathbb{U}$ in the quadratic ILC performance function J;

8) $G \in \mathbb{R}^{(p(L+1)) \times (d(L+1))}$ - intermittent matrix in ILC;

9) $\Gamma \in \mathbb{R}^{(d(L+1)) \times (p(L+1))}$ - learning rate to update $\mathbb{U}_{j+1}$;

10) $\bar{\mathbb{E}}_{i,m} \in \mathbb{R}^{K \times 1}$- modeling error vector in the m-th tuning within the i-th ILM *BATCH*;

11) $\Lambda_\mu \in \mathbb{R}^{1 \times K}, \Lambda_\sigma \in \mathbb{R}^{1 \times K}$ -learning rates vectors for μ and σ in ILM.