ILC-Based Fixed-Structure Controller Design for Output PDF Shaping in Stochastic Systems Using LMI Techniques

Hong Wang, *Senior Member, IEEE*, and Puya Afshar, *Member, IEEE*

***Abstract*—In this paper, a generalized state-space controller design for the shaping of the output probability density function (PDF) is presented for non-Gaussian dynamical stochastic systems. A Radial basis function (RBF) neural network is used to approximate the output PDF of the system. Such a neural network consists of a number of weights and corresponding basis functions. Using such an approximation, the dynamics of the original stochastic system can be expressed as the dynamics between the control input and the weights of the RBF neural network. The task of output PDF control can therefore be reduced to a RBF weight control together with an adaptive tuning of the basis function parameters (i.e., the centers and widths of the basis functions). To achieve this aim, the control horizon is divided into certain intervals hereinafter called *batches*. Using these definitions, the whole control strategy consists of three stages, namely (a) Sub-space parameter identification of the dynamic nonlinear model (that relates the control signal to the weights of the RBF neural network); (b) Weight tracking controller design using an LMI-based convex optimization technique; and (c) RBF basis functions shape tuning in terms of their centers and widths using an Iterative Learning Control (ILC) framework. Among the above stages, the first two are performed *within each batch*, while stage (c) is carried out *between any two adjacent batches*. Such an algorithm has the advantage of the batch-by-batch improvement of the closed-loop output PDF tracking performance. Moreover, the controller mentioned in stage (b) is a general controller in a state-space form. Stability analysis has been performed and simulation results are included to show the effectiveness of the proposed method, where encouraging results have been made.**

***Index Terms*—Convex optimization, generalized state-space controller, iterative learning control (ILC), output probability density function (PDF) control, sub-space system identification.**

I. Introduction

THE stochastic control problem has been of interest of many researchers during past three decades. Among existing stochastic control strategies, mean/variance and Linear Quadratic (LQ) control have been the first group of proposed methods ([1], [2]). Later on, stochastic adaptive and stochastic LQ martingale control were introduced ([3], [4]). Also, optimal and predictive stochastic control strategies have been developed for stochastic systems ([5], [6]), respectively. In more recent works, sliding mode control for jump stochastic systems [7], adaptive nonlinear stochastic control [8], and also robust fuzzy stochastic control [9] have also been proposed. In these methods, the system under control has been assumed to be of Gaussian type. For example, the random process has been considered as Markov and Wiener processes in [8] and [9], respectively.

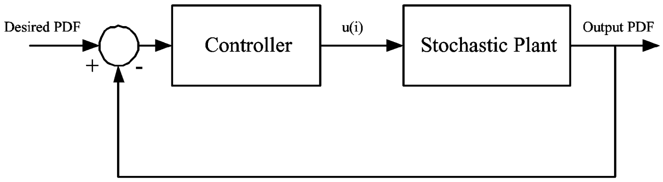

Fig. 1. Scheme of the output PDF control.

It has been shown that in systems where either the system variables or the noise are not Gaussian, existing methods may not be sufficient to characterize the closed loop system behavior. As a result, the control of output PDF, rather than the mean/vairance was proposed [10]. Fig. 1 shows the concept of output PDF control. Here, the set point is a desired PDF for the system output PDF to follow.

Since PDFs are non-negative and constrained functions, mathematical expressions of stochastic systems PDF generally lead to complicated partial differential equations (PDEs). For instance, for the stochastic systems described by Itô differential equations, the Fokker–Plack–Kolmogorov equation should be formulated in order to describe the dynamics of the system output PDF [11]. Furthermore, control of only the mean and variance of system outputs may not be sufficient for some practical applications. Examples are fibre length distribution control in paper-making [12], Molecular Weight Distribution Control (MWDC) ([13]–[16]), and Particle Size Distribution Control (PSDC) in polymerization and powder industries ([17]–[20] and [7]). For such practical control problems, the stochastic distribution control has been developed recently, where the purpose has been to design a controller so that the PDF of the system output follows a pre-specified desired distribution (represented by a targeted PDF), as close as possible. To simplify system modeling, B-Spline neural networks were initially used to approximate the output PDF and a number of algorithms and practical applications were developed ([10], [21], [22]). The key idea of the B-Spline approximation to the output PDF was to relate the control signal to the weights of the selected B-Spline neural network through a dynamical model. This transferred the problem of the output PDF tracking to the weights tracking problem. However, this group of methods

Manuscript received November 08, 2007; revised May 27, 2008. Current version published April 08, 2009. This work was supported by the UK Leverhulme Trust, the NSFC of China (60534010,60828007), and the 111 Project (B08015) with Northeastern University of China. Recommended by Associate Editor Z. Wang.

The authors are with the Control Systems Centre, University of Manchester, Manchester M60 1QD, U.K. (e-mail: hong.wang@.manchester.ac.uk).

Digital Object Identifier 10.1109/TAC.2009.2014934

0018-9286/$25.00 © 2009 IEEE

generally require a numerical optimization approach for the control signal determination. Moreover, numerical control design methods often suffer from two major disadvantages. First, the stability and closed loop performance are difficult to assess. Secondly, such controllers are often difficult to implement on-line due to their intensive computational load. In order to overcome these difficulties, fixed-structure controllers such as Proportional-Integral (PI) and Proportional-Integral-Derivative (PID) controllers have been applied to the output PDF control in our previous works ([23], [24]). It has been shown that a fixed-structure controller simplifies the closed-loop analysis and also guarantees the closed loop stability of the weight control system. However, in order to achieve a desired PDF tracking, a high dimensional time-domain model has to be used as a set of fixed span elements in the state space. This means that improving the closed loop tracking performance will be at a cost of excessive number of B-Spline basis functions. As a result, the idea of recently developed Iterative Learning Control (ILC) has been employed in our previous works ([25] and [26]) in order to improve the tracking performance. In this approach, the parameters of basis functions are used as tuning parameters of the ILC law and B-Spline basis functions are substituted for RBF basis functions to achieve a generalized expression of tuning parameters. In this regard, centers and widths of each RBF basis function have been treated as tuning parameters.

The objective of this paper is to replace the PI controller in [26] with the generalized fixed structure controller represented in a state-space form and then develop an ILC based algorithm so that the tracking performance of the output PDF can be improved along with the progress of batches. At the same time, the shape of basis functions will be automatically tuned to achieve a good fit of the RBF neural networks to the system output PDFs. In addition, it will be shown that while this generalized state-space controller can be used to control the output PDF shape with guaranteed closed loop stability, other controllers including state feedback and the PI controller in [26] can be regarded as special cases of the proposed controller.

This paper is organized as follows. A general framework of using ILC techniques to design the output PDF control will be described in Section II. In Section III, the problem of the output PDF control and related models (i.e., the RBF neural network and square-root PDF models along with the relevant dynamic models) will be introduced, where the general scheme of the ILC solution to the output PDF tracking control will be described. In Section IV, the details of the subspace parameter identification method will be given. Section V is comprised of the controller design procedure for both constrained and unconstrained output PDF models along with the conditions of the closed loop stability. Iterative Learning Control application and its relevant convergence analysis form the main subjects of the Section VI, while a set of simulation results of both constrained and unconstrained output PDF models will be presented in Section VII. Finally, concluding remarks will be given in Section VIII.

II. ILC-Based PDF Control Solution

Iterative Learning control (ILC) paradigm was first developed in 1984 for a class of systems where the closed-loop system operated similarly in a number of repetitive periods called iterations or batches [27]. Such systems are quite common in certain industries including chemical process control and manufacturing automation. The key idea is that the control input in the kth batch is determined based on the control input in the $(k-1)$th batch and a correction term which determines the ILC control type (e.g., P, D, etc). The correction term is related to the performance of the closed loop system in the $(k-1)$th batch as follows:

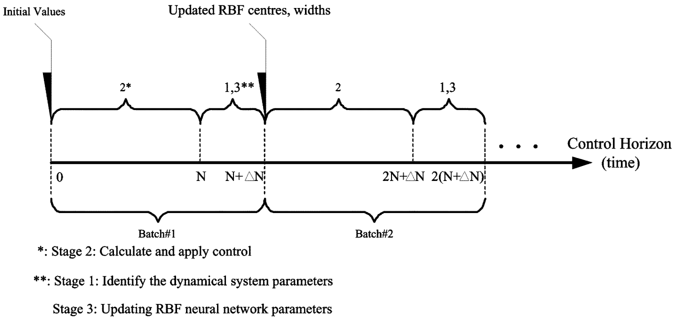
Initial Values
Updated RBF centres, widths
2*
1,3**
2
1,3
0
N
N+△N
2N+△N
2(N+△N)
Control Horizon (time)
Batch#1
Batch#2
*: Stage 2: Calculate and apply control
**: Stage 1: Identify the dynamical system parameters
Stage 3: Updating RBF neural network parameters

Fig. 2. ILC-based control design scheme.

$$u_k(i) = u_{k-1}(i) + \lambda H_{k-1}(i) \tag{1}$$

where i stands for the time instants satisfying $0 \leq i \leq m. m$ denotes the total number of time samples within a batch. $H(i)$ represents a function that is related to the tracking performance index (e.g., the difference between the desired output and manipulated variables in the system output), and λ is a learning rate which is chosen so that the iterative control law is convergent [28].

This strategy has the advantage of improving the closed-loop system tracking performance batch-by-batch ([29], [30] and [31]). In stochastic control research, the ILC approach has been applied to nonlinear stochastic systems for which the output tracking, rather than the output PDF tracking, has been developed [32]. Therefore, effort should be made so that the ILC can be applied in the design of the output PDF shaping control.

In order to apply the ILC approach in stochastic distribution control, the control horizon is first divided into a two main regimes, namely *within batches* and *between any two adjacent batches* as shown in Fig. 2.

As shown in Fig. 2, the control horizon is divided to a number of identical batches indexed by $k = (1, 2, \ldots)$ and these batches are specified by $[(k-1)(N+\Delta N), k(N+\Delta N)]$ where N is considered as the batch length, (during which the RBF parameters are fixed) and ΔN as the time period known as between adjacent batches. In this case the batch length N should be selected large enough so that the system almost reaches the steady state within each batch.

The process starts with setting initial values to the dynamic model parameters. Then within batches, i.e., $[(k-1)(N+\Delta N), kN+(k-1)\Delta N]$, fixed basis functions (see (4) of Section III) are used to generate the required control. The control signal is applied to the stochastic plant in such a way that the closed-loop system is stable. Then, between adjacent batches (i.e., during the time specified by $[kN+(k-1)\Delta N, k(N+\Delta N)]$) the RBF parameters (basis function centers and widths) are updated so that the measured output PDF gets closer to the desired PDF within the next batch.

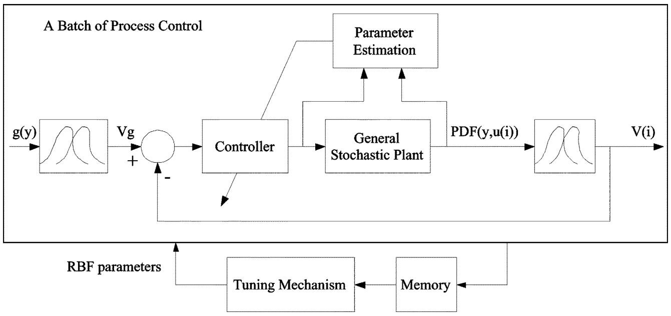

Fig. 3. Schematic diagram of the ILC-based PDF control.

Then a series of control signal and measured PDF signal are used to estimate the dynamical system parameters to be used within the next batch. During ΔN period of time the control law stays the same as that of $[0, N]$. This enables the tuning to be focused on the basis functions and the parameters of the weights dynamics. In this case, the structure of the proposed ILC-based PDF control approach with a general state-space controller will be as shown in Fig. 3. The three stages of the method proposed can be described as follows. First, the parameters of the dynamic model describing the output PDF are identified using a subspace system identification technique (for the first batch, initial values will be chosen). The identified parameters will be used within the next batch of control. Secondly, the general state-space controller is designed using LMI-based convex optimization technique so as to guarantee the closed loop Lyapunov stability. Finally, the stored data of the output PDFs from the previous batches are used to tune the parameters of the RBFs by using the ILC mechanism. Among the stages mentioned, the first two are carried out *within each batch* and the final stage is performed *between any two adjacent batches*, as previously illustrated in Fig. 2. This is different from existing ILC approaches where the control input is tuned directly between batches.

In this paper, all matrices have been assumed to have the compatible dimensions. In addition, subscripts i, j, and k stand for the time instance index of control, samples index of the output stochastic variable, and the number of the current batch of the system operation, respectively. Identity and zero matrices will be denoted by I and 0, respectively, and they are supposed to have compatible dimensions. Whenever used, the norm sign $\|.\|$ represents the Euclidean norm of the real vectors and matrices.

III. Problem Formulation

As described in Section I, the output PDF is approximated using a RBF neural network. As such, the PDF model representation will be expressed as follows. Both linear and square root models will be demonstrated.

A. *Model Representation*

Suppose that the random process $y \in [a, b]$ is the output of the stochastic system in Fig. 1. Let $u_k(i) \in \mathbb{R}^p$ be the control signal at the ith time instant within the kth batch of the control. Suppose that $u_k(i)$ controls the shape of the PDF of y and such a PDF is represented by $\gamma_k(y, u_k(i))$. By definition, the output PDF is expressed as follows:

$$P(a \le y < \zeta, u_k(i)) = \int_a^\zeta \gamma_k(y, u_k(i))dy \tag{2}$$

where $P(a \le y < \zeta, u_k(i))$ denotes the probability of y lying between a and ζ.

1) Dynamic Output PDF Modeling: Similar to the works in ([21]–[26]), we assume that the output PDF $\gamma_k(y, u_k(i))$ is continuous and bounded within each batch. Then the well-known RBF neural networks can be used to approximate the output PDF at the ith time instant within the kth batch as follows:

$$\gamma_k(y, u_k(i)) = \sum_{l=1}^{n-1} v_{l,k}(u_k(i))R_{l,k}(y) + h(V_k(i))R_{n,k}(y) \tag{3}$$

where $v_{l,k}(i)$ is the lth weight of the RBF neural network at the ith $(i = 1, 2, \ldots, m)$ sample time within the kth batch. Similar to [23], the RBF activation functions $R_{l,k}(y)$ are expressed as

$$R_{l,k}(y) = \exp\left(-\frac{(y_j - \mu_{l,k})^2}{2\sigma_{l,k}^2}\right), \quad (l = 1, 2, \ldots, n) \tag{4}$$

where $\mu_{l,k}$, $\sigma_{l,k}$ are the centers and widths of the RBF basis functions within the kth batch, respectively. These centers and widths will be tuned using iterative learning method between adjoint batches.

In addition, the output PDF is assumed to be measurable throughout the paper. This is mainly because the PDF signal is required for the control, as the weight elements are extracted based upon the measured output PDF. (see (12) for details). Practically, there are many systems where the output PDFs can be measured online. Examples are the paper-making systems and combustion systems where digital cameras are used online to take the images such as 2D solid distribution in paper web ([12], [21]–[24]) and flames distribution in combustion chambers, respectively. These images are then transferred into PDF information through online imaging processing. Also, fast PDF measurements are now available for particle size distribution measure in chemical engineering ([13]–[20]).

With these definitions, the dynamic model linking the output PDF and the RBF neural network weights vector can be expressed as follows [10]:

$$\begin{aligned} V_k(i+1) &= A_k V_k(i) + B_k u_k(i) + Ff(V_k(i)) \\ \gamma_k(y, u_k(i)) &= C_{0k}(y)V_k(i) + R_{n,k}(y)h(V_k(i)) \end{aligned} \tag{5}$$

where

$$\begin{aligned} V_k(i) &= [v_{1,k}(i), v_{2,k}(i), \ldots, v_{n-1,k}(i)]^T \\ C_{0k}(y) &= [R_{1,k}(y), R_{2,k}(y), \ldots, R_{n-1,k}(y)] \end{aligned}$$

are the weight vector and the basis functions row vector within the kth batch. The weight dynamical system (5) is considered as a linear model augmented with bounded state nonlinearities defined by function $f(V_k(i))$. In fact, the nonlinear bounded function $f(.)$ serves as noises and uncertainties and is assumed to satisfy the following Lipschitz condition:

$$\|f(V_{1,k}(i)) - f(V_{2,k}(i))\| \leq \|U(V_{1,k}(i) - V_{1,k}(i))\| \quad (6)$$

where U is a known matrix. In fact, term $h(V_k(i))$ in the right hand side of (3) is the nth RBF neural network weight separated from the first $n-1$. The reason for splitting the last element is that since the weights must satisfy the fundamental integral constraint of the probability (i.e., $\sum_{l=1}^{n} v_{l,k} \int_a^b R_{l,k}(y)dy = 1$), only the first $n-1$ elements in $V_k(i)$ vector are independent and thus the nth element can be obtained by using the first $n-1$ weights (as in (7) series of equations). Therefore, the nth weight element is considered as a function of other weight elements [10]. Such a dimension reduction also affects the representation of the output PDF as shown in (3).

Integrating the second equation in (5) over the output definition interval, the following can be verified:

$$h(V_k(i)) = a_{1,k}^{-1}(1 - V_k(i)a_{2,k}) \quad (7)$$

where it has been denoted that

$$a_{1,k} = \int_a^b R_{n,k}(y)dy, \quad a_{2,k} = \int_a^b C_{0k}(y)dy.$$

2) Square-Root Models of the Output PDF: Alternatively, square-root models can be used in order to avoid the negative values for the output PDF as addressed in [33] and [10]. The dynamic weight model will be the same as that used in (5) with the exception that the left hand side of the second equation will be changed to $\sqrt{\gamma_k(y, u_k(i))}$. However, this will add an additional constraint to the state vectors,(i.e., $V_k(i)$) so that the values of the weight vector remain non-imaginary subject to the PDF integral constraint (i.e., condition $\int_a^b \gamma_k(y, u_k(i))dy = 1$ must be satisfied within each batch). This implies that the resulting constraint in state vector will be as follows [10]

$$V_k^T(i)Q_{ab,k}V_k(i) \leq 1 \quad (8)$$

where

$$Q_{ab,k} = b_{1,k} - b_{3,k}^{-1}b_{2,k}^T b_{2,k}$$

in which

$$b_{1,k} = \int_a^b C_{0k}^T(y)C_{0k}(y)dy$$
$$b_{2,k} = \int_a^b R_{n,k}(y)C_{0k}(y)dy$$
$$b_{3,k} = \int_a^b R_{n,k}^2(y)dy.$$

It can be shown that $Q_{ab,k}$ is always a positive definite matrix. Also, the $h(V_k(i))$ function will take different form for the square root model ([10]). After describing the necessary mathematical tools for the development of the method, the details of the proposed three-stage design procedure will be described in Sections IV–VII.

IV. Sub-Space System Identification

Assuming that the output PDF and the control input for all previous sample times have been measured and the RBF parameters (i.e., the centers and the widths of the basis functions $R_{l,k}(y)$) have been updated after the completion of the $(k-1)$th batch, then in response to the tuning of basis functions, dynamic parameter matrices A_k and B_k should be identified using the recorded measurements of the control input and the output PDF before the kth batch starts.

From (5), it is obvious that the output equation is nonlinear. As a result, the subspace LTI system identification methods such as N4SID [34] and MOESP [35] cannot be applied directly. However, if a sequence of state vectors $V_{k-1}(i)$ can be calculated, the first equation in (5) would be a linear regression model with respect to input $u_{k-1}(i)$ and the relevant output vectors $V_{k-1}(i)$. Thus, while a sequence of $u_{k-1}(i)$ is measurable, vectors $[V_{k-1}^T(i) \quad h(V_{k-1}(i))]^T$ should be calculated.

Considering PDF model as discussed in (5) which is re-written in the following matrix form:

$$\gamma_k(y, u_k(i)) = [C_{0k}(y) \quad R_{n,k}(y)] \times \begin{bmatrix} V_{k-1}(i) \\ h(V_{k-1}(i)) \end{bmatrix}. \quad (9)$$

By pre-multiplying (9) with $[C_{0k}(y), R_{n,k}(y)]^T$ the following equality can be obtained:

$$\begin{bmatrix} C_{0k}^T(y) \\ R_{n,k}(y) \end{bmatrix} \gamma_k(y, u_k(i)) = \begin{bmatrix} C_{0k}^T(y)C_{0k}(y) & C_{0k}^T(y)R_{n,k}(y) \\ R_{n,k}(y)C_{0k}(y) & R_{n,k}^2(y) \end{bmatrix} \times \begin{bmatrix} V_{k-1}(i) \\ h(V_{k-1}(i)) \end{bmatrix}. \quad (10)$$

By performing integration on both sides of (10) over $[a,b]$, it can be seen that

$$\begin{bmatrix} \int_a^b C_{0k}^T(y)\gamma_k(y, u_k(i))dy \\ \int_a^b R_{n,k}(y)\gamma_k(y, u_k(i))dy \end{bmatrix} = \begin{bmatrix} b_{1,k} & b_{2,k}^T \\ b_{2,k} & b_{3,k} \end{bmatrix} \begin{bmatrix} V_{k-1}(i) \\ h(V_{k-1}(i)) \end{bmatrix}. \quad (11)$$

Hence, the weight vector within the $(k-1)$th batch can be determined by using the following operations based on the measured output PDF.

$$\begin{bmatrix} V_{k-1}(i) \\ h(V_{k-1}(i)) \end{bmatrix} = \begin{bmatrix} T_{1,k-1} & T_{2,k-1} \\ T_{2,k-1}^T & T_{3,k-1} \end{bmatrix} \begin{bmatrix} C_{2,k-1} \\ C_{3,k-1} \end{bmatrix} \quad (12)$$

where it has been denoted that

$$T_{1,k-1} = Q^{-1}b_{3,k}, \quad T_{2,k-1} = -Q^{-1}b_{2,k}$$
$$T_{3,k-1} = Q^{-1}b_{1,k}$$
$$Q^{-1} = \det \begin{bmatrix} b_{1,k} & b_{2,k}^T \\ b_{2,k} & b_{3,k} \end{bmatrix}$$
$$C_{2,k} = \int_a^b C_{0k}^T(y)\gamma_{k-1}(y, u_k(i))dy$$
$$C_{3,k} = \int_a^b R_{n,k}(y)\gamma_{k-1}(y, u_k(i))dy.$$

When using square root models, the above formulation is still valid by simply replacing $\gamma_k(y, u_k(i))$ with $\sqrt{\gamma_k(y, u_k(i))}$ in all the equations.

After the calculation of weight vectors, the well-known least squares method can be applied for the estimation of A_k and B_k using a sequence of V_{k-1}, u_{k-1}. For this purpose, the following linear regression model can be developed:

$$V_k(i) = \Theta_k^T \Phi_k(i-1) + E_k(i-1) \tag{13}$$

where

$$\begin{aligned}
&\Theta_k = \\
&\begin{bmatrix} \alpha_{1,1,k} & \cdots & \alpha_{1,n-1,k} & \beta_{1,1,k} & \cdots & \beta_{1,nu,k} \\ \vdots & & \vdots & \vdots & & \vdots \\ \alpha_{n-1,1,k} & \cdots & \alpha_{n-1,n-1,k} & \beta_{n-1,1,k} & \cdots & \beta_{n-1,nu,k} \end{bmatrix}^T \\
&E_k = [e_{1,k}, e_{2,k}, \ldots, e_{n-1,k}]^T \\
&\Phi_k(i-1) = \\
&[v_{1,k}(i-1), \ldots, v_{n,k}(i-1), u_{1,k}(i-1), \ldots, u_{nu,k}(i-1)]^T.
\end{aligned}$$

In the above equations, nu is the number of inputs, α, β and e stand for the corresponding elements in the parameter matrices A_k, B_k and the estimation error, respectively. Using (13), the Recursive Least Squares (RLS) algorithm can be employed to obtain matrices A_k and B_k for the kth batch.

By choosing a reasonable number of sample points y_j in the output PDF curve along its definition domain $[a, b]$ with ($j = 1, 2, \ldots, M$), the following recursive estimation can be applied for the identification of parameters in (13) in the same way as shown in [10]

$$\Theta_k(j+1) = \Theta_k(j) + \frac{P(j)\Phi_k(y_j, i-1)e_k(j)}{1+\Phi_k^T(y_j, i-1)P(j)\Phi_k(y_j, i-1)} \tag{14}$$

$$E_k(j) = V_k(i-1) - \Theta_k^T(j)\Phi_k(y_j, i-1) \tag{15}$$

$$P(j) = \left(I - \frac{P(j-1)\Phi_k(y_j, i-1)\Phi_k^T(y_j, i-1)}{1+\Phi_k^T(y_j, i-1)P(j-1)\Phi_k(y_j, i-1)}\right)P(j-1). \tag{16}$$

The initial value of the parameter matrix is selected as the last set values in the previous batch, and P is the matrix that is selected as big as possible initially (e.g., $P(1) = 1000I$ for the first batch).

V. Generalized Controller Design

As described in Section I, the following general state-space controller will be used to realize the closed loop control of output PDF:

$$\begin{aligned} x_{C,k}(i+1) &= K_k x_{C,k}(i) + L_k e_{s,k}(i) \\ u_k(i) &= M_k x_{C,k}(i) + N_k e_{s,k}(i) \end{aligned} \tag{17}$$

where $e_{s,k}(i) = V_g - V_k(i)$ stands for the tracking error. As mentioned before, the controller design procedure is commenced after the parameter estimation stage accomplished. The procedure is repeated once per batch. Corresponding to (5) and (17), the closed-loop system equation for the weight control loop in the kth batch can be written as

$$S_k(i+1) = \bar{A}_k S_k(i) + \bar{B}_k V_g + \bar{F} f(S_k(i)) \tag{18}$$

where

$$\begin{aligned}
S_k(i) &= \begin{bmatrix} V_k(i) \\ x_{C,k}(i) \end{bmatrix}, \quad f(S_k(i)) = \begin{bmatrix} f(V_k(i)) \\ 0 \end{bmatrix} \\
\bar{A}_k &= \begin{bmatrix} A_k - B_k N_k & B_k M_k \\ -L_k & K_k \end{bmatrix} \quad \bar{B}_k = \begin{bmatrix} B_k N_k \\ L_k \end{bmatrix} \\
\bar{F} &= \begin{bmatrix} F & 0 \\ 0 & 0 \end{bmatrix}.
\end{aligned}$$

It should be noted that when using the square root modeling method, the problem of controller design should be considered with the constraint $V^T(0) Q_{ab} V(0) \leq 1$. First, the case without state constraints is considered for non-square root models.

A. Un-Constrained Tracking Control Problem

Denoting

$$\tilde{A}_k = \begin{bmatrix} A_k & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{B}_k = \begin{bmatrix} 0 & B_k \\ I & 0 \end{bmatrix}, \quad \tilde{I} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \tag{19}$$

then the following theorem summaries the solvability conditions of the general state space form controller without state constraints.

Theorem 1: Within the kth batch and for any initial condition $S(0)$, if for some given constant λ the following linear matrix inequality (LMI)

$$\Psi_1 = \begin{bmatrix} -P_k & 0 & P_k^T \tilde{A}_k^T + R^T \tilde{B}_k^T & \lambda P_k \bar{U}_k^T \\ 0 & -\lambda^2 I & \bar{F}^T & 0 \\ \tilde{A}_k P_k + \tilde{B}_k R & \bar{F} & -P_k & 0 \\ \lambda \bar{U}_k P_k & 0 & 0 & -I \end{bmatrix} < 0. \tag{20}$$

is solvable for some positive definite matrix P_k and matrix R, then

1) the closed loop system (18) is stable;

2) $\lim_{i \to \infty} e_{s,k}(i) = 0$;

and the controller parameters can be calculated using

$$\begin{bmatrix} K_k & L_k \\ M_k & N_k \end{bmatrix} = RP^{-1}\tilde{I}^{-1}. \tag{21}$$

Proof: To prove the stability and tracking performance of the closed-loop system (18), let us consider the following Lyapunov function candidate:

$$\begin{aligned} L(S_k(i), i) = {} & S_k^T(i) P_k^{-1} S_k(i) \\ & + \sum_{j=1}^{i-1} \left[\|\lambda U V_k(i)\|^2 - \|\lambda f(V_k(i))\|^2 \right]. \end{aligned} \tag{22}$$

Differentiating the (22) over the time gives

$$\Delta_L(i) = L(S_k(i+1), i+1) - L(S_k(i), i)$$

$$\begin{aligned} &= \tilde{S}_k^T(i)\Psi_{2,k}\tilde{S}_k(i) + 2\tilde{S}_k^T(i)\bar{A}_k^T P_k^{-1}\bar{B}_k V_g \\ &\quad + 2f^T(V_k(i))\bar{F}^T P_k^{-1}\bar{B}_k V_g \\ &\quad + V_g^T \bar{B}_k^T P_k^{-1}\bar{B}_k V_g < 0 \end{aligned} \tag{23}$$

where

$$\Psi_{2,k} = \begin{bmatrix} \bar{A}_k^T P_k^{-1}\bar{A} - P_k^{-1} + \lambda^2\bar{U}^T\bar{U} & \bar{A}_k^T P_k^{-1}\bar{F} \\ \bar{F}^T P_k^{-T}\bar{A}_k & \bar{F}^T P_k^{-1}\bar{F} - \lambda^2 I \end{bmatrix}.$$

Using the well-known Schur complement formula, (23) can be further reduced to

$$\Psi_{3,k} = \begin{bmatrix} -P_k^{-1} + \lambda^2\bar{U}^T\bar{U} & 0 & \bar{A}_k^T \\ 0 & -\lambda^2 I & \bar{F}^T \\ \bar{A}_k & \bar{F} & -P_k \end{bmatrix} < 0.$$

By pre-multiplying $\Psi_{3,k}$ with $diag(P_k^T, I, I)$ and post multiplying it with $diag(P_k, I, I)$ and also applying Schur complement, the necessary condition for stability will be as follows:

$$\Psi_{4,k} = \begin{bmatrix} -P_k & 0 & P_k^T\bar{A}_k^T & \lambda P_k\bar{U}_k^T \\ 0 & -\lambda^2 I & \bar{F}^T & 0 \\ \bar{A}_k P_k & \bar{F} & -P_k & 0 \\ \lambda\bar{U}_k P_k & 0 & 0 & -I \end{bmatrix} < 0. \tag{24}$$

Finally, by substituting matrices $\tilde{A}$, $\tilde{B}$, and $\tilde{I}$ into (24), $\Psi_{1,k}$ can be obtained.

If (20) holds, a positive scalar δ exists so that $\Psi_{2,k} \leq -\delta I$. Along with (18) it can be verified that

$$\begin{aligned} \Delta_L(i) \leq &-\delta\|\tilde{S}_k\|^2 + 2\|\bar{A}_k^T P_k^{-1}\bar{B}_k V_g\|\|\tilde{S}_k\| \\ &+ 2f^T(V_k(i))\bar{F}^T P_k^{-1}\bar{B}_k V_g \\ &+ V_g^T\bar{B}_k^T P_k^{-1}\bar{B}_k V_g. \end{aligned} \tag{25}$$

It is obvious that the right-hand side of the inequality (25) is a second degree polynomial with respect to $\|\tilde{S}_k(i)\|$. Thus, denoting $m = 2f^T(V_k(i))\bar{F}^T P_k^{-1}\bar{B}_k V_g$ and $n = V_g^T\bar{B}_k^T P_k^{-1}\bar{B}_k V_g$, it can be shown that $\Delta_L(i) \leq 0$ holds if

$$\begin{aligned} \|\tilde{S}_k(i)\| \geq \delta^{-1}\Big(&\|\bar{A}_k^T P_k^{-1}\bar{B}_k V_g\| \\ &+ \sqrt{\left\|\bar{A}_k^T P_k^{-1}\bar{B}_k V_g\right\|^2 + \delta(m+n)}\Big) \end{aligned} \tag{26}$$

which implies (27), shown at the bottom of the page. This means that the stability of the closed-loop system can be confirmed. ■

To discuss the system tracking performance, suppose two different trajectories $\psi_1(i)$ and $\psi_2(i)$ on (18) corresponding to fixed initial conditions. Also consider the V_g to be an input. Then the difference of two trajectories can be expressed as

$$\epsilon(i) = \psi_1(i) - \psi_2(i)$$

$$\epsilon(1) = 0. \tag{28}$$

It can be shown that variable $\epsilon(i)$ satisfies the following dynamic equation:

$$\epsilon(i+1) = \bar{A}_k\epsilon(i) + \bar{F}[f(\psi_1) - f(\psi_2)]. \tag{29}$$

Similar to the developments of the closed loop system, the Lyapunov function can be re-written as

$$\begin{aligned} L(\epsilon(i), \psi_1(i), \psi_1(i), i) &= \epsilon^T(i)P^{-1}\epsilon(i) \\ &+ \sum_{j=1}^{i-1}\left[\|\lambda U\epsilon(i)\|^2 - \|\lambda(f(\psi_1(i)) - f(\psi_1(i)))\|^2\right] \end{aligned} \tag{30}$$

with the similar difference equation as (23)

$$\Delta_L(i) = \bar{\epsilon}^T(i)\Phi_3\bar{\epsilon}(i) < -\delta\|\bar{\epsilon}(i)\|^2 \tag{31}$$

where $\bar{\epsilon}(i) = \left[\epsilon^T(i), f(\psi_1) - f(\psi_2)\right]^T$. As a result, the closed loop system is asymptotically stable around $\epsilon = 0$ neighborhood, which means that the tracking performance of the system has been achieved. After exploring the solvability conditions, the feasible design procedure can be provided.

In addition, it is obvious that the previously proposed controllers such as PI controller proposed in [26] can be regarded as a special case of the general state-space form controller, as represented in this work. Indeed, by fixing $L = -I$ and $K = 0$, the resulting controller is a PI controller. Moreover, by setting $K = L = N = 0$ the resulting controller is a state-feedback one.

B. Constrained General State Space Controller Design

In order to solve the control design problem for the square root model of the output PDF, constraint (8) should be satisfied. Based upon theorem 1, it can be concluded that if δ is sufficiently large, then the constraint (8) can be guaranteed automatically. However, analysis on conditions under which the constraints on partial states can be guaranteed are provided in the following theorem.

Theorem 2: For initial conditions $S_k(0)$ satisfying constraint (8), if the LMI (20) holds and the following linear constraint

$$2\|\bar{B}_k V_g\| \leq \lambda_{\min}(P_k) \tag{32}$$

is satisfied for a positive δ; then within each batch:

- the closed-loop system (18) is exponentially stable;
- tracking error converges to zero, i.e., $\lim_{i\to\infty} e_{s,k}(i) = 0$;
- state constraint $V_k^T(i)Q_{ab}V_k(i) \leq 1$ is satisfied;

and the generalized state space controller can be calculated from (21). In (32), $\lambda_{\min}(P_k)$ stands for the smallest eigenvalue of the matrix P_k.

$$\left\|\tilde{S}_k(i)\right\| \leq \max\left\{\left\|\tilde{S}_k(1)\right\|, \delta^{-1}\left(\left\|\bar{A}_k^T P_k\bar{B}_k V_g\right\| + \sqrt{\left\|\bar{A}_k^T P_k\bar{B}_k V_g\right\|^2 + \delta(m+n)}\right)\right\} \tag{27}$$

Proof: Using the Lyapunov function chosen as (22), and denoting (18), it can be verified that

$$\begin{aligned}\Delta_L(i) = {}& \tilde{S}_k^T(i)\left(\bar{A}_k^T P_k^{-1}\bar{A}_k - P_k^{-1}\right)\tilde{S}_k(i)\\ &+2\tilde{S}_k^T(i)\bar{A}_k^T P_k^{-1}\bar{B}_k V_g + V_g^T\bar{B}_k^T P_k^{-1}\bar{B}_k V_g. \end{aligned}\tag{33}$$

Since $\left(\bar{A}_k^T P_k^{-1}\bar{A}_k - P_k^{-1}\right) < 0$ is guaranteed by (20), there must exist a positive δ so that $\left(\bar{A}_k^T P_k^{-1}\bar{A}_k - P_k^{-1}\right) < -\delta^{-1}I$. Thus, it can be verified that

$$\begin{aligned}\Delta_L(i) \le {}& -\delta^{-1}\|\tilde{S}_k(i)\|^2 + 2\tilde{S}_k^T(i)\bar{A}_k^T P_k^{-1}\bar{B}_k V_g\\ &+ V_g^T\bar{B}_k^T P_k^{-1}\bar{B}_k V_g + 2f^T(V_k(i))\bar{F}^T P_k^{-1}\bar{B}_k V_g\\ \le {}& -\delta^{-1}\|\tilde{S}_k(i)\|^2 - \tilde{S}_k^T(i)\bar{A}_k^T P_k^{-1}\bar{A}_k\tilde{S}_k(i)\\ &+ 2V_g^T\bar{A}_k^T P_k^{-1}\bar{B}_k V_g + 2f^T(V_k(i))\bar{F}^T P_k^{-1}\bar{B}_k V_g\\ \le {}& -\delta^{-1}\|\tilde{S}_k(i)\|^2\\ &- \left[P_k^{-1/2}\bar{A}_k\tilde{S}_k(i) - P_k^{-1/2}\bar{B}_k V_g\right]^T\\ &\times\left[P_k^{-1/2}\bar{A}_k\tilde{S}_k(i) - P_k^{-1/2}\bar{B}_k V_g\right]\\ &+ 2V_g^T\bar{A}_k^T P_k^{-1}\bar{B}_k V_g + 2f^T(V_k(i))\bar{F}^T P_k^{-1}\bar{B}_k V_g\\ \le {}& -\delta^{-1}\|\tilde{S}_k(i)\|^2 + \rho \end{aligned}\tag{34}$$

where $\rho = 2\lambda_{\max}(P_k^{-1})\|\bar{B}_k V_g\|$. Then it can be shown that $\Delta_L(i) < 0$ holds if $\delta\rho < \|\tilde{S}_k(i)\|^2$. In order to satisfy the state constraint expressed in (8), it can be shown that

$$\begin{aligned}V_k^T(i)Q_{ab}V_k(i) <{}& \|V_k(i)\|^2\|Q_{ab}\|\\ \le{}& \|\tilde{S}_k(i)\|^2\|Q_{ab}\|\\ \le{}& \rho\delta\|Q_{ab}\| \le 1. \end{aligned}\tag{35}$$

■

The feasible design procedure for the controller can be summarized as follows.

- Before the beginning of a batch, ρ should be selected as a given constant.
- Select δ and apply it to the calculation procedure (20).
- Repeat the stages for the controller design without state constraints.

VI. Iterative Learning Control Application

Once the $(k-1)$th batch is completed, the centers and widths of the radial basis functions are updated so that the closed-loop system tracking performance is improved for the kth batch. This forms the main content of this section.

A. Tuning of Radial Basis Functions

For the Radial Basis Functions defined by (4), the following P-type ILC law is used for the tuning of the parameters in the basis functions (4) between any two adjacent batches, namely the $(k-1)$th and the kth batches

$$\begin{aligned}\mu_{l,k} &= \mu_{l,k-1} + \Lambda_\mu E_{k-1}\\ \sigma_{l,k} &= \sigma_{l,k-1} + \Lambda_\sigma E_{k-1}\end{aligned}\tag{36}$$

where the performance indices of the $(k-1)$th batch are defined as follows:

$$E_{k-1} = [J_{k-1}(1), J_{k-1}(2), \dots, J_{k-1}(m)]^T \tag{37}$$

while m stands for the total number of time instants within a batch. The development of the tuning rules will depend on the model based on which the output PDF has been modeled. As such, two formulations should be made, one for the unconstrained model and the other for the square root model. For the unconstrained model of the output PDF, the J in (37) can be expressed as

$$J_{k-1}(i) = \int_a^b (\gamma_k(y, u_k(i)) - g(y))^2 dy \tag{38}$$

which is the performance at the ith sampling instant of the $(k-1)$th batch. It is desired that the measured output PDF $\gamma_k(y, u_k(i))$ will get closer to the desired PDF $g(y)$ as the batches k pass, i.e., the measured PDF is matched the desired PDF as $k \to \infty$. Furthermore, the learning parameters in (36) are defined as

$$\begin{aligned}\Lambda_\mu &= \alpha_\mu[\lambda_1, \lambda_2, \dots, \lambda_m]\\ \Lambda_\sigma &= \alpha_\sigma[\lambda'_1, \lambda'_2, \dots, \lambda'_m]\end{aligned}\tag{39}$$

where λ and λ' are the learning elements and α_μ and α_σ are the learning rates to be determined. From (37), it is obvious that all the elements in E_{k-1} are nonnegative, allowing either positive or negative values for the learning rates. This means that centres and the width of the radial basis functions can be varied either to increase or to decrease along the progress of batches. For the square root models, (38) is modified to read

$$J_{k-1}(i) = \int_a^b \left(\sqrt{\gamma_k(y, u_k(i))} - \sqrt{g(y)}\right)^2 dy \tag{40}$$

while the tuning rules for the basis functions are the same as those given in (36).

B. Convergence Analysis

The iterative learning algorithm (36) takes place between $(k-1)$th batch and the kth batch. It involves two learning rate parameter vectors as given in (39). It is therefore important to ensure that these learning vectors are properly selected to guarantee the convergence of the updating of the basis functions between two batches. The key issue is that such tuning vectors should guarantee the improvement of the closed loop performance batch by batch. This can be achieved if the following condition is satisfied:

$$\frac{F_k}{F_{k-1}} = \frac{\sum\limits_{i=1}^{m} J_k(i)}{\sum\limits_{i=1}^{m} J_{k-1}(i)} \le 1 \tag{41}$$

where

$$F_k = \sum_{i=1}^{m} J_k(i) \tag{42}$$

is the measure of the overall closed loop performance within the kth batch. Since $J_k(i)$ is non-negative, condition (41) would mean that

$$\Delta F_k = F_k - F_{k-1} \le 0. \tag{43}$$

In the following, convergence conditions will be formulated for both non-constrained and the square-root models.

1) Convergence Conditions for Unconstrained PDF Models: Similar to the discussions in [25], the convergence conditions for unconstrained PDF models can be expressed as (44), shown at the bottom of the next page, where the learning parameter vectors should be selected so that the above inequality holds.

2) Convergence Conditions for Square-Root PDF Models: Discussion for the convergence conditions in square root models have been provided in [26]. It can be summarized as follows:

$$\sum_{i=1}^{m}\int_{a}^{b}\left[2\left(\sqrt{\gamma_{k-1,i}(y)}-\sqrt{g(y)}\right)\Delta\sqrt{\gamma_{k-1,i}(y)}\right]dy \le 0 \tag{45}$$

together with

$$\begin{aligned}\Delta R_{l,k-1}(y) = {} & \frac{y_j-\mu_{k-1}}{\sigma_{k-1}^2}R_{l,k-1}(y)\Lambda_\mu E_{k-1} \\ & + \frac{(y_j-\mu_{k-1})^2}{\sigma_{k-1}^3}R_{l,k-1}(y)\Lambda_\sigma E_{k-1}\end{aligned} \tag{46}$$

and

$$\begin{aligned}\Delta\sqrt{\gamma_{k-1,i}(y)} &= \sqrt{\gamma_{k-1,i}(y)}-\sqrt{\gamma_{k-2,i}(y)} \\ &= \sum_{l=1}^{n}V_l(i)\Delta R_{l,k-1}(y)\end{aligned} \tag{47}$$

where

$$\begin{aligned}\Delta R_{l,k-1}(y) &= R_{l,k-1}(y)-R_{l,k-2}(y) \\ &\approx \frac{\partial}{\partial\mu_{k-1}}R_{l,k-1}(y)\Delta\mu_k \\ &\quad + \frac{\partial}{\partial\sigma_{k-1}}R_{l,k-1}(y)\Delta\sigma_k\end{aligned} \tag{48}$$

with notations $\Delta\mu_k = \mu_k - \mu_{k-1}$ and $\Delta\sigma_k = \sigma_k - \sigma_{k-1}$. As a result, the sufficient conditions for the convergence are summarized as follows.

Iterative tuning rule (36) should be applied with the learning rates satisfying (44) when the unconstrained PDF models are used, and with (45) when the square-root models are used. There are also some constraints on the convergence as listed in the following.

1) Both μ and σ, must be non-negative;
2) Within a batch, $J_k(i+1)$ is more important than $J_k(i)$. Thus, the weight of $J_k(i+1)$ in (39) should be more than $J_k(i)$. This means that $[\lambda_1, \lambda_2, \ldots, \lambda_m]$ and $[\lambda_1{}', \lambda_2', \ldots, \lambda_m']$ should be both gradually increasing vectors within the batch.

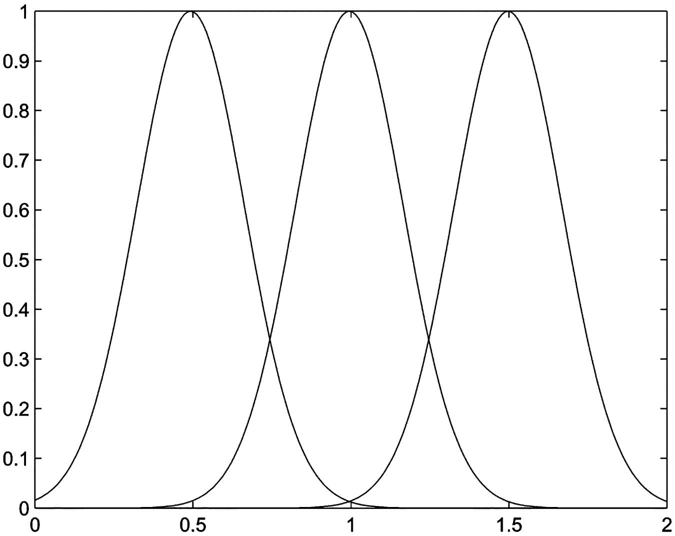

Fig. 4. Initial distribution of RBFs.

VII. Simulation Results

In this section, a simulation study of the proposed method will be described. First, the system model and RBF neural network components are introduced and then the performance of the ILC-based PDF shaping system based on constrained and unconstrained models will be investigated.

A. Stochastic System Model and Problem Statement

It is assumed that the stochastic system output variable is distributed over interval $[0, 2]$, i.e., $y \in [0, 2]$. In order to approximate the output PDF, a three-layer neural network with three radial basis activation functions is used. Fig. 4 shows the initial formation of the RBFs. According to (4), the initial parameters of activation functions in Fig. 4 are chosen as follows:

$$\mu_1 = 0.5, \mu_2 = 1.0, \quad \mu_3 = 1.5, \quad \sigma_1 = \sigma_2 = \sigma_3 = 0.2.$$

This would mean that the output PDF of the stochastic system will be described as

$$\begin{gathered}\gamma_k(y, u_k(i)) = C_{0k}(y)V_k(i) + R_3(y)h(V_k(i)) \\ \text{where} \\ V_k(i) = [v_{1,k}(i), v_{2,k}(i)]^T \\ C_{0k}(y) = [R_{1,k}(y), R_{2,k}(y)].\end{gathered}$$

$$V(i)[V(i)R_k(y) - V_gR_g(y)](y-\mu_k) \times \left[\Lambda_\mu E_{k-1} + \frac{y-\mu_k}{\sigma_k}\Lambda_\sigma E_{k-1}\right] < 0 \tag{44}$$

For the case of constrained PDF model, term $\gamma_k(y, u_k(i))$ is replaced by $\sqrt{\gamma_k(y, u_k(i))}$. The weight vector behaves dynamically as expressed in (5) with the following parameters:

$$A = \begin{bmatrix} 0.20 & 0.40 \\ 0.41 & 0.81 \end{bmatrix}, \quad B = \begin{bmatrix} 0.01 & 0.01 \\ 0.01 & 0.02 \end{bmatrix}$$

$$\text{and}$$

$$F = \begin{bmatrix} 0.02 & 0.00 \\ 0.00 & 0.01 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},$$

$$f = \begin{bmatrix} \sin(2\pi v_1(i)) \\ \sin(2\pi v_2(i)) \end{bmatrix}.$$

The initial value of the weight vector is set as $V_1(0) = [0 \quad 0]^T$. The problem is to find a state space controller to make the output PDF of the stochastic system follow a desired distribution as closely as possible. This means that within each batch, parameter matrices K, L, M, and N must be found so that the closed loop system is internally stable. In addition, between any two adjacent batches, the RBF parameters $\mu_{k,i}$, $\sigma_{k,i}$, $(i = 1, 2, 3)$ and also system parameter matrices A_k and B_k must be trained so that the ILC cost function (42) takes monotonically decreasing values along with the progress of batches. The simulation study begins with the unconstrained PDF model as follows.

B. Results of Unconstrained PDF Model

In this section, all the design stages with regards to the unconstrained PDF model will be provided. First, the specific problem requirements such as set points will be introduced and then each of the three design stages studied before will be described separately in detail.

1) Desired PDF Settings: It is desired that the measured output PDF follows a distribution as described by RBF basis functions parameters (i.e., centers and width) defined as follows:

$$\mu_{g1} = 0.2, \quad \mu_{g2} = 0.8, \quad \mu_{g3} = 1.2,$$
$$\sigma_{g1} = \sigma_{g2} = \sigma_{g3} = 0.1$$

In addition, for the unconstrained mode of PDF modeling, the desired dynamical weights are set to $V_g = [0.01 \quad 0.02]^T$ which results in the dependent weight (calculated by (7)) to be $h(V_g) = 0.104$. Furthermore, 200 samples defined for the output variable and 50 for the time samples within each batch $(i = 1, 2, \ldots, 50)$. Such parameters will control the desired unconstrained PDF shape as shown in Fig. 5. Hence, it is desired that the resulting controlled output PDF have the same shape as of shown in Fig. 5. In addition to above mentioned parameters, it must be noted that the total number of 5 batches $(k = 1, 2, \ldots, 5)$ have been chosen for the case of unconstrained control whereas four batches have been considered for constrained PDF tracking control.

2) Subspace Parameter Estimation: As mentioned before, the first batch begins with the initial values chosen for the dynamical system parameters A and B. Initially, the following values are chosen for above mentioned matrices:

$$A_1 = \begin{bmatrix} 0.80 & 0.35 \\ 0.74 & 0.85 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0.2 & -0.1 \\ 0.1 & 0.3 \end{bmatrix}.$$

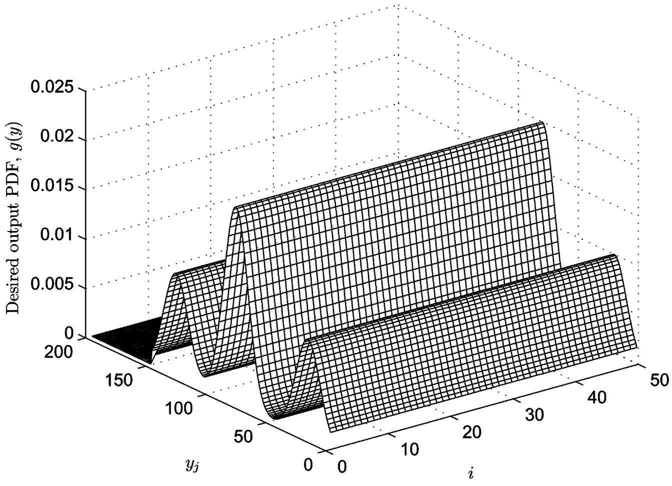

Fig. 5. Desired output PDF.

Also, the initial values to be used in parameter identification algorithms expressed by (14)–(16) are set to $\theta_0 = 0$ and $P(0) = 1000I$. It is worth noting that based on the values stored within the first batch, the parameter estimation system yields the following parameter values to be used within the second batch:

$$A_2 = \begin{bmatrix} 0.1996 & 0.3909 \\ 0.3968 & 0.7871 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0.0135 & 0.0114 \\ 0.0174 & 0.0130 \end{bmatrix}.$$

After the end of the fourth batch, the parameter identification algorithm results in the following values for parameter matrices:

$$A_5 = \begin{bmatrix} 0.2024 & 0.3979 \\ 0.4039 & 0.7980 \end{bmatrix}, \quad B_5 = \begin{bmatrix} 0.0085 & 0.0098 \\ 0.0092 & 0.0110 \end{bmatrix}.$$

Having set the parameter matrix values, the next step is to calculate the control signal to be applied to the system.

3) Control Signal Calculation: The parameters of the general state-space controller (17) can be found by solving a feasibility problem given by (20) and (21). The controller is initially set to $x_C(0) = [0 \quad 0]^T$ and the MATLAB LMI toolbox is used to solve the problem to obtain the following controller parameters for use within the first batch:

$$K_1 = \begin{bmatrix} 0.0068 & 0.0068 \\ 0.0068 & 0.0068 \end{bmatrix}, \quad L_1 = \begin{bmatrix} -0.0267 & 0.0034 \\ -0.0267 & 0.0034 \end{bmatrix}$$

with

$$M_1 = \begin{bmatrix} -0.0146 & -0.0146 \\ -0.0293 & -0.0293 \end{bmatrix}$$
$$N_1 = \begin{bmatrix} -5.1853 & -2.2370 \\ -2.2706 & -0.9740 \end{bmatrix}.$$

After having measured the weight vector $V_1(i)$, the dependent weight $h(V_1(i))$ is calculated by (7) resulting the output PDF take a shape as shown in Fig. 6. This means that the PDF tracking performance within the initial batch of operation will be as shown in Fig. 7. It can be seen that although the closed loop system is internally stable, the dependent weight has taken some negative values which is due to the application of the

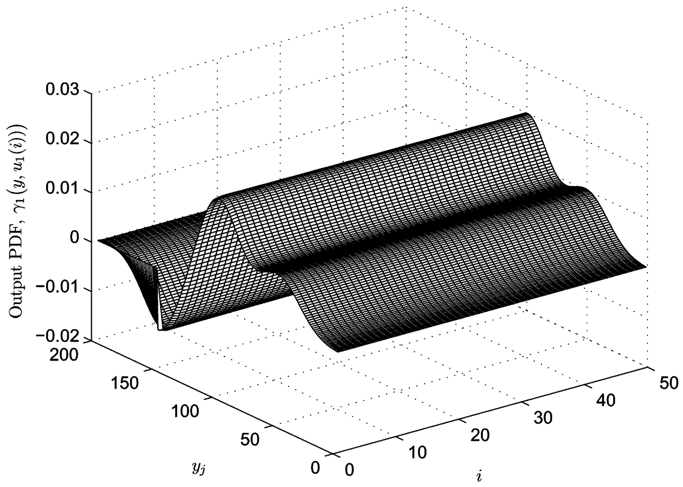

Fig. 6. Measured Output PDF after the first batch.

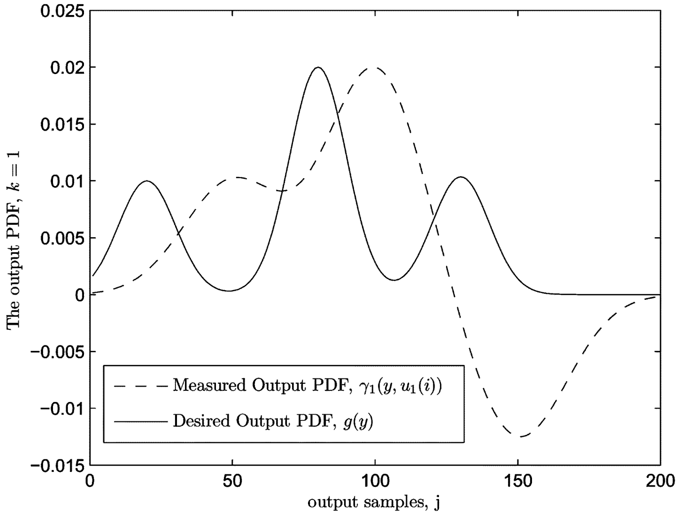

Fig. 7. Tracking of desired PDF in the end of first batch.

linear PDF models. In total, five batches are considered. After the final batch, the parameters of the controller are as follows:

$$K_5 = \begin{bmatrix} 0.0053 & 0.0053 \\ 0.0053 & 0.0053 \end{bmatrix}, \quad L_5 = \begin{bmatrix} -0.0178 & -0.009 \\ -0.0178 & -0.009 \end{bmatrix}$$

with

$$M_5 = \begin{bmatrix} 0.1579 & 0.1579 \\ -0.1339 & -0.1339 \end{bmatrix}, \quad N_5 = \begin{bmatrix} -106.64 & -0.20 \\ 89.82 & -0.23 \end{bmatrix}.$$

As such, the performance of the inner loop (i.e., the weight control loop) can be illustrated in Fig. 8. Indeed, Fig. 8 implies that the solution to the corresponding LMI feasibility problem results in an efficient weight tracking and ensures the stability of the inner control loop. Using these parameters, the response of the output PDF resulted after the final batch is shown in Fig. 9. Also, the PDF tracking performance in the last batch of the operation is shown in Fig. 10 which represents a satisfactory tracking performance within the last batch. The connecting point between figures Figs. 6,7 and 9, 10 is the ILC-based parameter tuning presented in Section VII-B-IV.

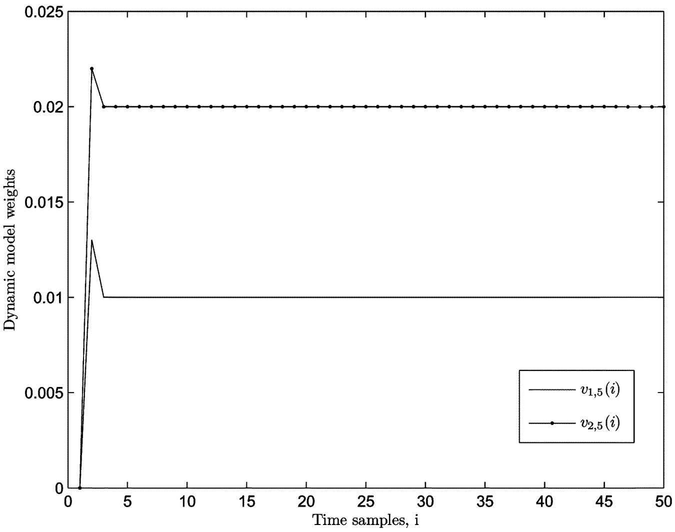

Fig. 8. Weights of the RBF neural network as the end of final batch.

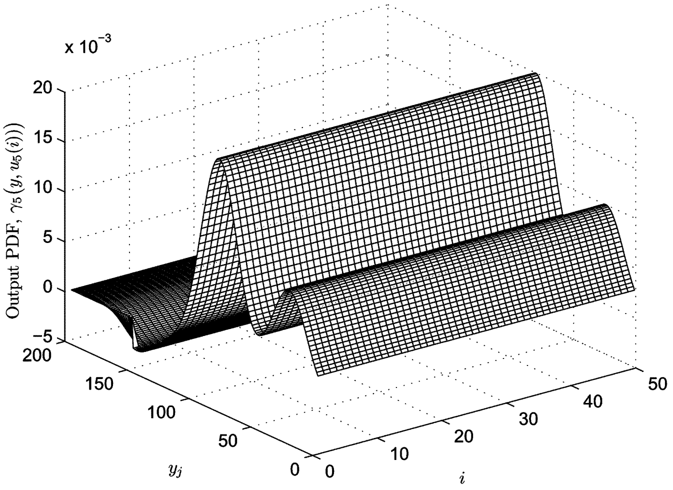

Fig. 9. Output PDF in the end of final batch.

4) ILC-Based Parameter Tuning: For the purpose of ILC-based RBF parameter tuning previously expressed by (36), the following learning rate values have been chosen based on (39):

$$\Lambda_\mu = \begin{bmatrix} -50 & 0 & 0 \\ 0 & -75 & 0 \\ 0 & 0 & -18 \end{bmatrix} \begin{bmatrix} 0.06 \times j \\ 0.03 \times j \\ 0.02 \times j \end{bmatrix}$$

and

$$\Lambda_\sigma = \begin{bmatrix} -30 & 0 & 0 \\ 0 & -30 & 0 \\ 0 & 0 & -30 \end{bmatrix} \begin{bmatrix} 0.1 \times j \\ 0.1 \times j \\ 0.1 \times j \end{bmatrix}, \quad j = 0, 1, \ldots, 50.$$

After the final batch is completed, the values for the RBF centers and width are given by $\mu_1 = 0.2204$, $\mu_2 = 0.7956$, $\mu_3 = 1.3607$ and $\sigma_1 = \sigma_2 = \sigma_3 = 0.1127$. These result in the tracking performance as shown in Fig. 10. As mentioned before, for ILC-based PDF control efficiency, the cost function (42) must be assessed for having monotonically decreasing values. Fig. 11 illustrates the shape of the ILC performance function which implies the convergence of the algorithm after the final batch. However, as it can be observed from the above figures, the output PDF has taken negative values when it comes to the

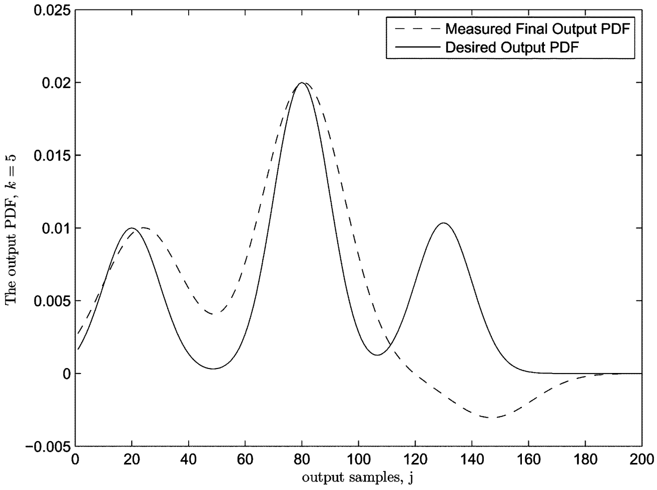

Fig. 10. Tracking of desired PDF in the end of final batch.

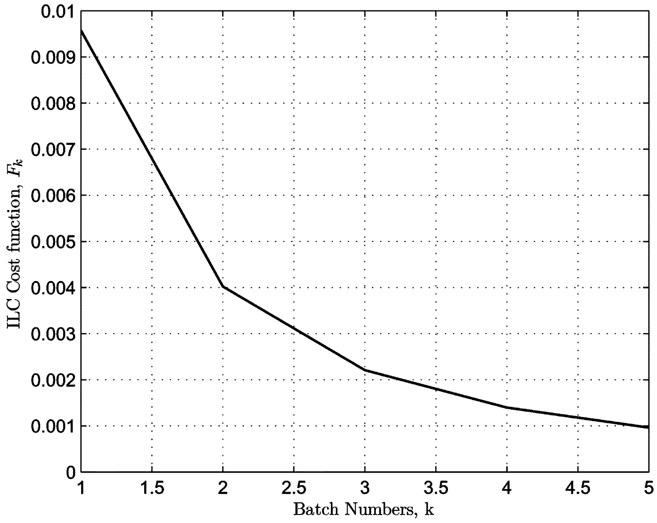

Fig. 11. ILC Performance.

dependent weight. To prevent this numerical phenomena, the square-root model will be used for the output PDF approximation.

C. Performance of the Square-Root PDF Models

It has been shown that the square root models guarantee the positiveness of the output PDF at the cost of bringing a new constraint to the state vector of the system. As a result, the constraint should be satisfied throughout the different batches.

1) Desired Square-Rooted PDF Settings: All initial parameters are the same as the unconstrained model except that the desired weights are set to $V_g = [0.07 \quad 0.05]^T$, $h(V_g) = 0.2160$. Also, 20 time samples are used instead of 50. As such, the desired output PDF shape will be as shown in Fig. 12. Similar to unconstrained design, subspace system identification is considered here.

2) Subspace Parameter Identification-Constrained Model: Starting with the same initial values as the unconstrained

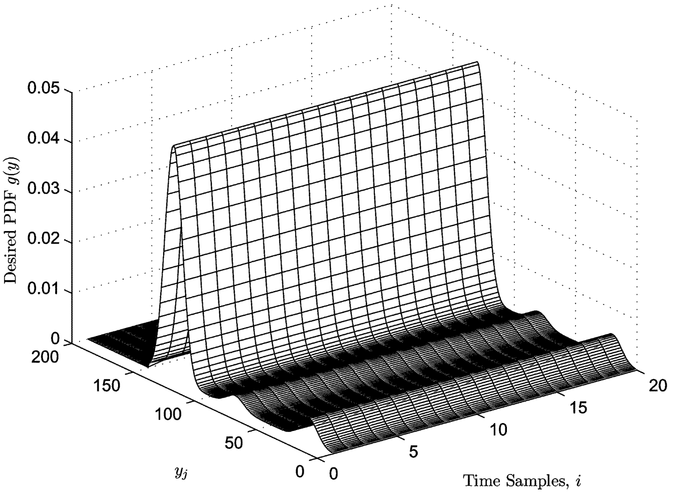

Fig. 12. Desired output PDF.

model, the subspace parameter estimation results in the following dynamical system matrix values to be used within the second batch:

$$A_2 = \begin{bmatrix} 0.6591 & 0.4694 \\ 0.4706 & 0.3358 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -0.1080 & -0.0379 \\ -0.0819 & -0.0272 \end{bmatrix}.$$

Also, in the final batch the system matrices A_4 and B_4 are identified to have the following values:

$$A_4 = \begin{bmatrix} 0.6902 & 0.4337 \\ 0.4928 & 0.3102 \end{bmatrix}, \quad B_4 = \begin{bmatrix} -0.2308 & -0.0041 \\ -0.1704 & -0.0030 \end{bmatrix}.$$

3) Constrained Control Signal Calculation: Similar to the unconstrained model, controller parameter matrices will be calculated based on (20) and (21) whereas in constrained model the constraint (32) must be considered. Also, an initial value $V_1(0) = [0.01 \quad 0.01]^T$ has been considered for the weight vector. The above mentioned equations yield the following controller parameters within the first batch:

$$K_1 = \begin{bmatrix} -0.0024 & -0.0024 \\ -0.0024 & -0.0024 \end{bmatrix}$$
$$L_1 = \begin{bmatrix} -0.0353 & 0.0274 \\ -0.0353 & 0.0274 \end{bmatrix}$$
and
$$M_1 = \begin{bmatrix} -0.0078 & -0.0078 \\ -0.0157 & -0.0157 \end{bmatrix}$$
$$N_1 = \begin{bmatrix} -4.6028 & -2.6236 \\ -1.2056 & -1.7472 \end{bmatrix}.$$

By measuring weigh vector $V_1(i)$ and calculating the dependent weight $h(V_1(i))$ at the end of the first batch, the output PDF will be shaped as displayed in Fig. 13. The PDF tracking performance within the first batch shows the difference between the desired and measured PDFs clearly in Fig. 14. After passing through the remaining four batches, the control parameters in the final batch are given by

$$K_4 = \begin{bmatrix} -0.0057 & -0.0057 \\ -0.0057 & -0.0057 \end{bmatrix}$$

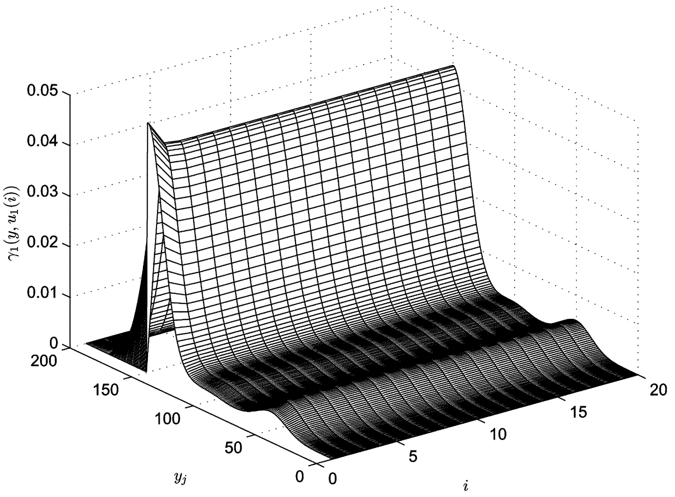

Fig. 13. Measured output square-root PDF in the end of first batch.

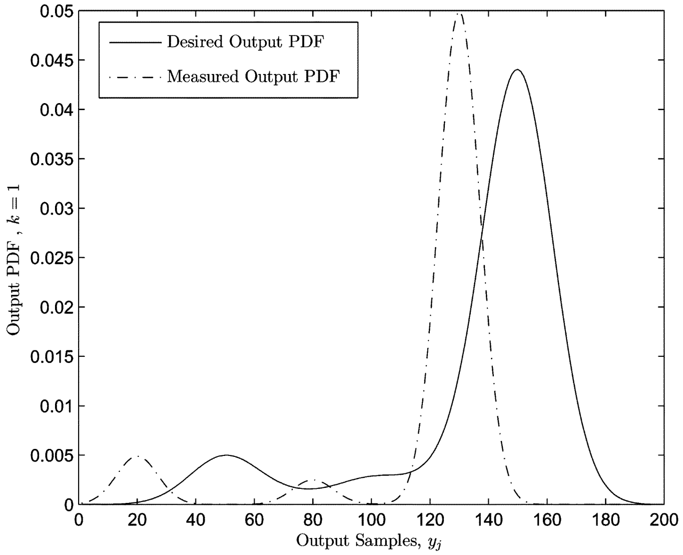

Fig. 14. Tracking of desired PDF in the end of first batch.

$$L_4 = \begin{bmatrix} 0.0248 & -0.0044 \\ 0.0248 & -0.0044 \end{bmatrix}$$

and

$$M_4 = \begin{bmatrix} -0.001274 & -0.001274 \\ -0.004037 & -0.004037 \end{bmatrix}$$

$$N_4 = \begin{bmatrix} 111.86 & 0.27 \\ 353.21 & -0.9 \end{bmatrix}.$$

Based upon the designed controller, the weight control performance is illustrated in Fig. 15, which shows that a satisfactory control for the weights of the neural network has been obtained. The controlled values of the weights should be employed in forming the PDF values within the final batch. The above mentioned values are used to calculate the final output PDF for the system. Fig. 16 shows the output PDF of the square root model within the final batch of the ILC. It can be seen from Fig. 17, the output PDF tracking performance shows a significant improvement in comparison to the performance within the first batch, as shown in Fig. 7.

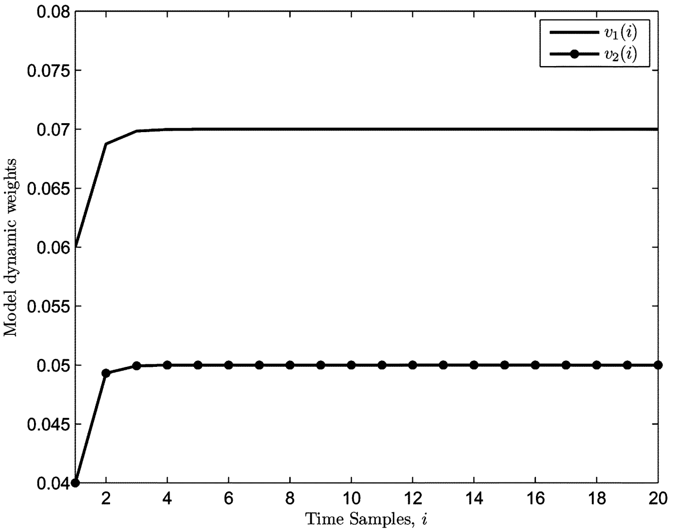

Fig. 15. Weights of the RBF neural network in final batch.

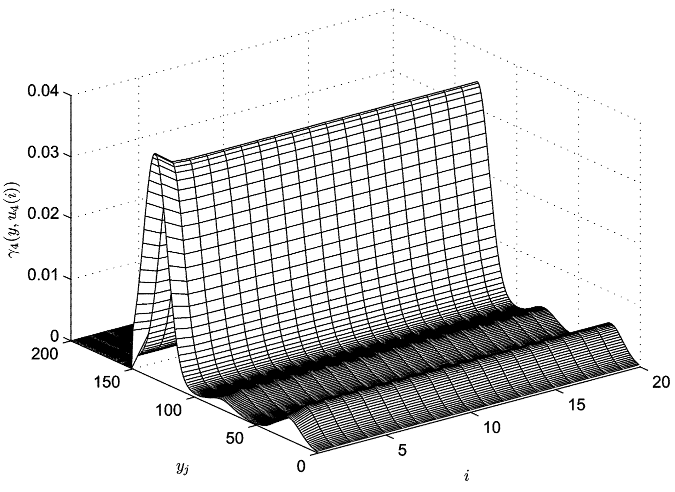

Fig. 16. Measured output PDF in the end of final batch.

4) Constrained ILC-Based Parameter Tuning: The learning rates corresponding to the constrained PDF tracking problem are chosen as

$$\Lambda_\mu = \begin{bmatrix} -8.60 & 0.00 & 0.00 \\ 0.00 & -14.50 & 0.00 \\ 0.00 & 0.00 & -18.0 \end{bmatrix} \begin{bmatrix} 0.0030 \times j \\ 0.0015 \times j \\ 0.0015 \times j \end{bmatrix}$$

along with

$$\Lambda_\sigma = \begin{bmatrix} -0.67 & 0.000 & 0.000 \\ 0.000 & -0.67 & 0.000 \\ 0.000 & 0.000 & -0.67 \end{bmatrix} \begin{bmatrix} 0.005 \times j \\ 0.005 \times j \\ 0.005 \times j \end{bmatrix}$$

where $j = 0, 1, \ldots, 20$. After the final batch is accomplished, the tuned final parameters of the RBF neural network are obtained as $\mu_1 = 0.23$, $\mu_2 = 0.76$, $\mu_3 = 1.4$, and $\sigma_1 = \sigma_2 = \sigma_3 = 0.14$. Finally, the ILC performance function, which is supposed to be monotonically decreasing along with the advances of the batches, is shown in Fig. 18. Compared to the unconstrained control results, the output PDF tracking performance

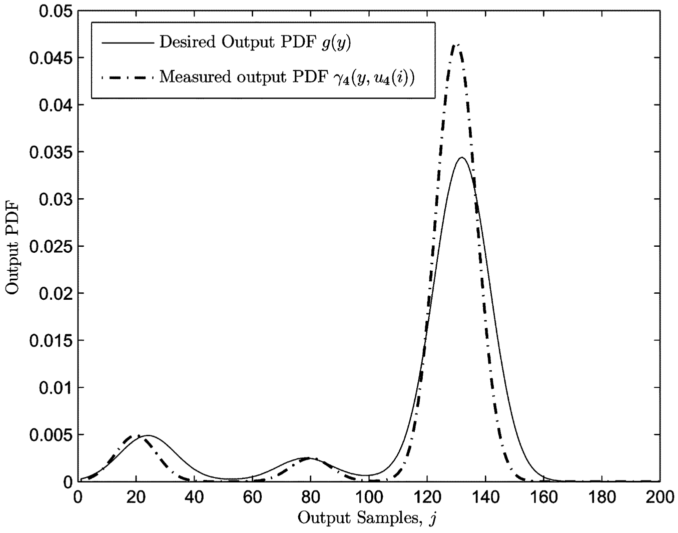

Fig. 17. Square root PDF tracking in the end of final batch.

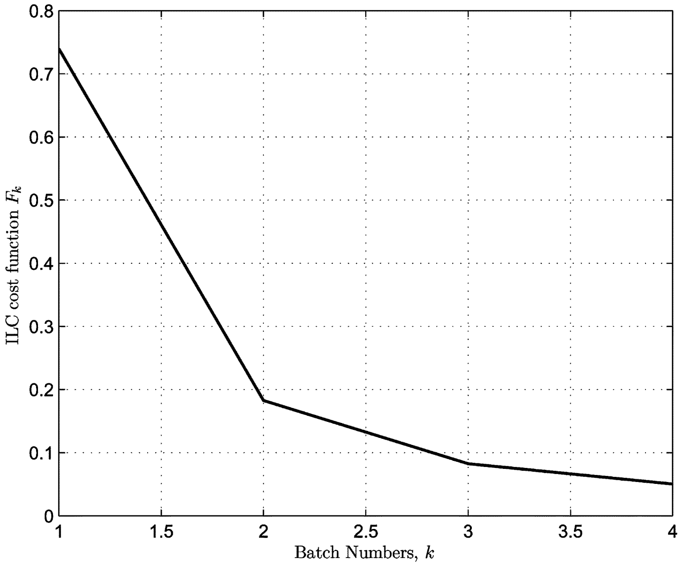

Fig. 18. Performance function of the ILC law in square root PDF model.

under square root constraint represents a better tracking performance over shorter length of process operation (i.e., fewer number of batches). Also, the square root model guarantees the positiveness of the measured output PDF at all control instances.

VIII. Conclusion

A generalized state-space form of controller has been developed for the output PDF control of general stochastic systems. While the method reduces the problem of the PDF tracking to the problem of neural network weight tracking, it separates the control design into two important stages: one carried out in the space domain and the other in the time domain. At the time domain stage, an LMI approach has been applied in order to guarantee the closed-loop stability and also to find the parameters of the generalized state-space controller. At the space domain stage, a P-type ILC law was employed to tune the parameters of the RBF neural network. This has resulted in a performance improvement of the closed loop system along with the advances of batches. Using the proposed time domain controller, the previously developed methods such as the PI controller [26], or the state feedback controllers can be regarded as special cases of the proposed controller. The method also provides a feasible solution for the cases where the ordinary PDF modeling leads to negative values for the output PDF.

References

[1] K. J. Astrom, *Introduction to Stochastic Control Theory*. New York: Academic, 1970.

[2] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*. Englewood Cliffs, NJ: Prentice Hall, 1971.

[3] V. Solo, “Stochastic adaptive control and martingale lmit theory,” *IEEE Trans. Automat. Control*, vol. 35, no. 1, pp. 66–71, Jan. 1991.

[4] M. H. C. Everdij and H. A. P. Bolm, “Embedding adaptive JLQG into LQ martingale control with a completely observable stochastic control matrix,” *IEEE Trans. Automat. Control*, vol. 41, no. 3, pp. 424–430, Mar. 1996.

[5] E. Yaz, “Optimal stochastic control for performance and stability robustness,” *IEEE Trans. Automat. Control*, vol. 38, no. , pp. 757–760, May 1993.

[6] N. M. Filatov and H. Unbehausen, “Adaptive preditive control policy for nonlinear stochastic systems,” *IEEE Trans. Automat. Syst.*, vol. 40, no. 11, pp. 1943–1949, Nov. 1995.

[7] S. Shi, N. H. El-Farra, M. Li, and P. Mhaskar, “Predictive control of particle size distribution in particulate processes,” *Chem. Eng. Sci.*, vol. 61, no. 1, pp. 268–281, Jan. 2006.

[8] H. B. Ji and H. S. Xi, “Adaptive output-feedback tracking of stochastic nonlinear systems,” *IEEE Trans. Automat. Control*, vol. 51, no. 2, pp. 355–360, Feb. 2006.

[9] W. H. Wing and K. Y. Cai, “Model-independent robust stabilization for uncertain markovian jump nonlinear systems via fuzzy control,” *IEEE Trans. Automat. Control*, vol. 36, no. 6, pp. 509–519, Jun. 2006.

[10] H. Wang, *Bounded Dynamic Stochastic Systems: Modeling and Control*. New York: Springer-Verlag, 2000.

[11] H. Risken, *The Fokker—Planck Equation: Methods of Solution and Applications*. New York: Springer-Verlag, 1989.

[12] G. A. Smook, *Handbook for Pulp and Paper Technology*. Vancouver, BC, Canada: Angus Wilde Publication, Inc., 1992.

[13] T. J. Crowley and K. Y. Choi, “Experimental studies on optimal molecular weight distribution control in a batch-free radical polymerization process,” *Chem. Eng. Sci.*, vol. 53, no. 15, pp. 2769–2790, Aug. 1998.

[14] H. S. Cho, J. S. Chung, and W. Y. Lee, “Control of molecular weight distribution for polyethylene catalyzed over Ziegler–Natta/Metallocene hybrid and mixed catalysts,” *J. Molecular Catalysis A: Chem.*, vol. 159, no. 2, pp. 203–213, Oct. 2000.

[15] M. Vicente, C. Sayer, J. R. Leiza, G. Arzamendi, E. L. Lima, J. C. Pinto, and J. M. Asua, “Dynamic optimization of non-linear emulsion copolymerization systems: Open-loop control of composition and molecular weight distribution,” *Chem. Eng. J.*, vol. 85, no. 2,3, pp. 339–349, Jan. 2002.

[16] Y. Shibasaki, T. Araki, R. Nagahata, and M. Ueda, “Control of molecular weight distribution in poly-condensation polymers 2. poly(ether ketone) synthesis,” *Eur. Polymer J.*, vol. 41, no. 10, pp. 2428–2433, Oct. 2005.

[17] C. A. Dunbar and A. J. Hickey, “Evaluation of probability density functions to approximate particle size distribution of representative pharmaceutical aerosols,” *J. Aerosol Sci.*, vol. 31, no. 7, pp. 813–831, Jul. 2000.

[18] 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 polymerization,” *Comput. Chem. Eng.*, vol. 27, no. 8,9, pp. 1153–1163, Sep. 2003.

[19] C. D. Immanuel and F. J. Doyle, III, “Open-loop control of particle size distribution in semi-batch emulsion copolymerization using a genetic algorithm,” *Chem. Eng. Sci.*, vol. 57, no. 20, pp. 4415–4427, Oct. 2002.

[20] K. Lee, J. H. Lee, D. R. Yang, and A. W. Mahoney, “Integrated run-to-run and on-line model-based control of particle size distribution for a semi-batch precipitation reactor,” *Comput. Chem. Eng.*, vol. 26, no. 7, 8, pp. 1117–1131, Aug. 2002.

[21] H. Wang, “Robust control of the output probability density functions for multivariable stochastic systems,” *IEEE Trans. Automat. Control*, vol. 44, no. 11, pp. 2103–2107, Nov. 1999.

[22] H. Wang and J. Zhang, "Bounded stochastic distribution control of pseudo ARMAX systems," *IEEE Trans. Automat. Control*, vol. 46, no. 3, pp. 486–490, Mar. 2001.

[23] L. Guo and H. Wang, "Applying constrained nonlinear generalized pi strategy to PDF tracking control through square root B -spline models," *Int. J. Control*, vol. 77, no. 17, pp. 1481–1492, 2004.

[24] L. Guo and H. Wang, "PID controller design for output PDFs of stochastic systems using linear matrix inequalities," *IEEE Trans. Syst., Man, Cybern. B*, vol. 35, no. 1, pp. 65–71, Feb. 2005.

[25] H. Wang and P. Afshar, "Radial basis function based iterative learning control for stochastic distribution system," in *Proc. IEEE Int. Symp. Intell. Syst.*, Munich, Germany, 2006, pp. 100–105.

[26] H. Wang and P. Afshar, "ILC -based generalized PI control of output PDF of stochastic systems using LMI and RBF neural networks," in *Proc. IEEE Conf. Decision Control*, San Diego, CA, 2006, pp. 5048–5053.

[27] A. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering Operation of Robots by Learning," *J. Robot. Syst.*, vol. 1, pp. 123–140, 1984.

[28] Z. Bien and K. M. Huh, "High-order iterative learning control algorithm," *Proc. Inst. Elect. Eng. D*, vol. 136, pp. 105–112, 1989.

[29] C. Ham, Z. H. Qu, and J. Kaloust, "A new framework of learning control for a class of nonlinear systems," in *Proc. Amer. Control Conf.*, 1995, pp. 3024–3028.

[30] D. H. Hwang, Z. Bien, and S. R. Oh, "Iterative learning control method for discrete-time dynamic systems," *Proc. Inst. Elect. Eng. D*, vol. 138, no. 2, pp. 139–144, 1991.

[31] S. R. Oh, Z. Bien, and I. H. Suh, "An iterative learning control method with application for the robot manipulator," *IEEE J. Robot. Automat.*, vol. 4, no. 5, pp. 508–514, Apr. 1998.

[32] H. F. Chen and H. T. Feng, "Output tracking for nonlinear stochastic systems by iterative learning control," *IEEE Trans. Automat. Control*, vol. 49, no. 4, pp. 583–588, Apr. 2004.

[33] C. T. J. Dodson and J. Scharcanski, "Information geometric similarity measurement for near-random stochastic processes," *IEEE Trans. Syst., Man, Cybern. A*, vol. 33, no. 4, pp. 435–440, Jul. 2003.

[34] P. Van Overschee and D. De Moor, "N4SID: Subspace algorithm for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, no. 1, pp. 75–93, 1994.

[35] Verhaegen, "Identification of deterministic part of MIMO state space models given in innovationfrom input-output data," *Automatica*, vol. 30, no. 1, pp. 61–74, 1994.

Hong Wang (M'97–SM'05) was born in Beijing, China, in 1960. He received the B.S. degree from Huainan University of Mining Engineering, Huainan, China, in 1982, and the M.S. and Ph.D. degrees from Huazhong University of Science and Technology (HUST), Huazhong, China, in 1984 and 1987, respectively.

He was a Research Fellow at Salford, Brunel, and Southampton Universities (UK) before joining University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K., in 1992, and has been a Professor in process control since 2002. He is a Visiting Professor at the Northeastern University (China), HUST, and the Institute of Automation, Chinese Academy of Sciences. He now serves as an Editorial Board Member for the *Journal of Measurement and Control*, the *Transactions of the Institute of Measurement and Control*, the *IMechE Journal of Systems and Control Engineering*, *Automatica Sinica*, and the *Journal of Control Theory and Applications*. He has published over 200 papers and three books in the following areas. His research interests are stochastic distribution control, fault detection and diagnosis, nonlinear control, and data based modeling for complex systems.

Dr. Wang was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He is a Fellow of IET and InstMC.

Puya Afshar (S'06–M'08) was born in Tehran, Iran, in 1980. He received the B.S. degree from K.N.Toosi University of Technology (KNTU), Tehran, Iran, in 2002, the M.S. degrees from Tarbiat Modares (TMU), Tehran, Iran, in 2004, and the Ph.D. degree from The University of Manchester Control Systems Centre, Manchester, U.K., in 2007.

Since 2008, he has been a Research Associate in advanced control systems. His research interests are stochastic distribution control, iterative learning control, agent based automation systems, intelligent control, and active fault tolerant control systems.