

An ILC-Based Adaptive Control for General Stochastic Systems With Strictly Decreasing Entropy

Puya Afshar, *Member, IEEE*, Hong Wang, *Senior Member, IEEE*, and Tianyou Chai, *Fellow, IEEE*

Abstract—In this paper, a new method for adaptive control of general nonlinear and non-Gaussian unknown stochastic systems has been proposed. The method applies the minimum entropy control scheme to decrease the closed-loop randomness of the output under an iterative learning control (ILC) basis. Both modeling and control of the plant are performed using dynamic neural networks. For this purpose, the whole control horizon is divided into a certain number of time domain subintervals called *batches* and a pseudo-D-type ILC law is employed to train the plant model and controller parameters so that the entropy of the closed-loop tracking error is made to decrease batch by batch. The method has the advantage of decreasing the output uncertainty versus the advances of batches along the time horizon. The analysis on the proposed ILC convergence is made and a set of demonstrable experiment results is also provided to show the effectiveness of the obtained control algorithm, where encouraging results have been obtained.

Index Terms—Adaptive controller, entropy, iterative learning control (ILC), neural networks, stochastic systems and optimization.

I. INTRODUCTION

RESEARCH on stochastic control has been of a great importance in the past decades. This is due to the fact that most of the practical industrial systems are subjected to random noises and uncertainties. Among various existing approaches of the stochastic control, the minimum variance control introduced in [1] is still one of the key methods in this research field, where the key issue is to minimize the uncertainties of the closed-loop stochastic systems. Other works proposed have considered a variety of methods ranging from linear quadratic martingale control [2] and predictive stochastic control [3] to sliding mode control for jump stochastic systems [4], adaptive nonlinear stochastic control [5], and robust fuzzy control for uncertain Markovian stochastic systems [6]. In most of the methods mentioned above, the tracking error has played a vital role for the assessment of the closed-loop performance.

Manuscript received May 26, 2007; revised March 26, 2008 and August 20, 2008; accepted November 07, 2008. Current version published February 27, 2009. This work was supported by the U.K. Leverhulme Trust (F/00 120/BC), China National Science Foundation under Grants 60534010, 60828007, and 60821063, National Basic Research Program of China under Grant 2009CB320601, and the 111 project (B08015).

P. Afshar and H. Wang are with the Control Systems Centre, School of Electrical and Electronic Engineering, The University of Manchester, M60 1QD, Manchester, The United Kingdom (e-mail: hong.wang@manchester.ac.uk).

T. Chai is with the Research Center of Automation, Northeastern University, Shenyang 110004, China.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2008.2010351

Most of the existing works have assumed that the random signals interfering with the system are Gaussian. However, for systems subjected to nonsymmetrically distributed random noises, the spread area of the noise distribution cannot be described precisely by only using its variance. In this context, second-order moments of the tracking error signals cannot fully characterize the performance of the closed-loop control system in terms of the randomness measure. As such, a new measure of uncertainty, namely, the entropy of the tracking error of the closed-loop system, should be employed for the closed-loop control design of the non-Gaussian stochastic systems. In fact, it can be shown that when the noise is Gaussian, the minimum entropy control is equivalent to minimum variance control [7], [8].

Entropy originated in the information theory and the economical studies. There are different interpretations of the entropy. However, it was first introduced by Shannon [9] as a measure of uncertainty, and later on, as the average information content on randomness for a given probability density function [10], [11]. The main advantage of this definition is that it provides a generalized randomness measure of a random variable by measuring its dispersion rather than its mean or variance. With a simple inspiration from the information theory, it can be said that a control system is similar to a classical information learning system, which is a scheme based on interaction between pairs of information particles. The goal in such a system is to maximize the mutual information between the actual input, i.e., the process measured value in a control system, and the desired output, which is equivalent to the set point in control systems. Such a definition is literally equivalent to the minimization of the tracking error entropy for closed-loop stochastic systems.

Among different entropy measure definitions, Shannon's and Renyi's entropy measures have the same minima. However, the α -order Renyi's quadratic entropy, which will be used throughout this paper, has an advantage of computational efficiency over Shannon's entropy definition. It can also be shown that the well-known mean square error criterion widely used in system identification is a special case of the Renyi's mutual information maximization criterion. Renyi's entropy measure for a random variable x can be expressed as follows:

$$H(x) = \frac{1}{1-\alpha} \log \left(\int \gamma^\alpha(x) dx \right) \quad (1)$$

where γ stands for the probability density function (pdf) of the random variable x . The integration is performed over the definition range of x , wherever the pdf is nonzero.

As mentioned above, the entropy links the control and estimation to the information theory. Indeed, entropy measure has

been applied to some of the control and estimation problems before. In earlier works, the entropy formulation provides an information-theoretic interpretation of the feedback process [12]. Later on, a unified framework was presented for approximation, optimization, and adaptive control using the entropy associated with the uncertainty for controller design [13]. Moreover, the entropy minimization has been used as an optimization index to estimate the states of dynamic systems and several feedback control strategies have been established using the entropy measure [14]. In stochastic systems control, a robust optimal controller has been designed for uncertain stochastic systems with partial information, where the uncertainty has been described as a constraint on the relative entropy between the nominal and perturbed noise distributions [15]. More recently, the minimum entropy control for stochastic systems has been developed as reported in [7], [16], and [17], which are basically different from the minimum-entropy-based H_∞ control [18]. As mentioned in [8], the original definition of the entropy used in H_∞ control for linear time invariant systems at $s = \infty$ is defined in terms of the closed-loop transfer function as follows:

$$I_d(G; l; \infty) = -\frac{l^2}{2\pi} \int_{-\infty}^{+\infty} \ln |\det(I - l^{-2} G^*(j\omega)G(j\omega))| d\omega. \quad (2)$$

In the above definition, the transfer function G is supposed to be strictly proper satisfying $\|G\|_\infty < l$, G^* denotes that the adjoint of G , and I_d is finite. By the entropy measure definition in this work, the mutual information maximization idea is replaced in the probabilistic point of view, which is used to quantify the uncertainty of a general stochastic control system. However, in almost all previous works, the plant model and also the noise distribution functions have been assumed known.

The entropy control problem is in fact an application of stochastic distribution control, known as pdf control [19]. In the output pdf control methodologies, the task of control is to find a deterministic control input so that the output pdf follows a prespecified desired pdf. In a similar way, the task of entropy control is to find a deterministic control signal such that the closed-loop tracking error pdf gets as close to a narrowly shaped normal distribution as possible. Having said that, no matter what application of pdf control is dealt with, the pdf signal needs to be estimated. As mentioned in [20], existing methods applied to decouple the pdf output and control input will lead to a high-dimensional time span requiring a large number of basis functions to ensure the accuracy of the pdf estimation. In addition, pdf signal will be available only after a certain number of time samples are passed. As a solution, the idea of iterative learning control (ILC)-based design has been applied to overcome the dimensionality problem and also to improve the closed-loop output pdf tracking (or closed-loop output entropy) performance periodically. Such a solution can be considerably useful for industrial applications where the feedback signal cannot be measured online, due to instrumentation restrictions, such a molecular weight distribution (MWD) control in polymerization reaction control problem [21].

In this paper, an adaptive method based on ILC theory has been proposed for the non-Gaussian systems with unknown nonlinear dynamics and noise distribution. The key idea is to divide the control horizon into certain number of time-domain intervals hereinafter called *batches* and separate the design procedure into two main categories, *within each batch* and *between any two adjacent batches*. Based on the definitions made, the following design stages are used in this paper:

- 1) approximating the unknown plant Jacobian using the identified neural network model of the plant;
- 2) calculating the control input;
- 3) updating the controller parameters using an ILC idea and returning to 1)

The second stage above will be carried out within each batch and the other stages are performed between any two adjacent batches.

To realize such a design, the control horizon is divided to K identical batches specified as $[(k-1)(N+\Delta N), k(N+\Delta N)]$ ($k = 1, 2, \dots, K$) where N is considered as the batch length, during which the parameters are fixed and ΔN as the time period known as *between adjacent batches*. Within each batch, i.e., $[(k-1)(N+\Delta N), kN + (k-1)\Delta N]$, fixed basis functions are used to generate the required control. Then, between adjacent batches, i.e., during the time specified by $[kN + (k-1)\Delta N, k(N+\Delta N)]$, the model and controller parameters are updated by minimum entropy criteria. 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. The above design approach is illustrated below as in Fig. 1.

This paper is organized as follows. In Section II, the problem of the ILC-based minimum entropy control is introduced. Section III consists of the details of controller design and the nonlinear system identification using neural networks. The ILC convergence analysis is described in Section IV, while the application of proposed method to a test rig together with experimental results are described in Section V. Finally, concluding remarks are made in Section VI.

II. PROBLEM FORMULATION

It is assumed that the stochastic plant to be controlled can be expressed as the following nonlinear ARMAX model:

$$y_k(i) = f_p(y_k(i-1), y_k(i-2), \dots, y_k(i-n_a), u_k(i-n_d), u_k(i-n_d-1), \dots, u_k(i-n_b-n_d+1), \omega_k(i)) \quad (3)$$

where f_p is the unknown nonlinear plant equation, which is assumed continuous, bounded, and first-order differentiable with respect to all of its variables. As indicated in Fig. 2, $y_k(i)$ and $u_k(i)$ are the i th samples of the measured output and the control signal within the k th batch, respectively. Also, n_a , n_b , and n_d are the structural orders of the system, respectively. Furthermore, ω_k is the bounded non-Gaussian random noise with an unknown pdf $\gamma_\omega(x)$, where x is assumed to be defined on a known interval $[a, b]$.

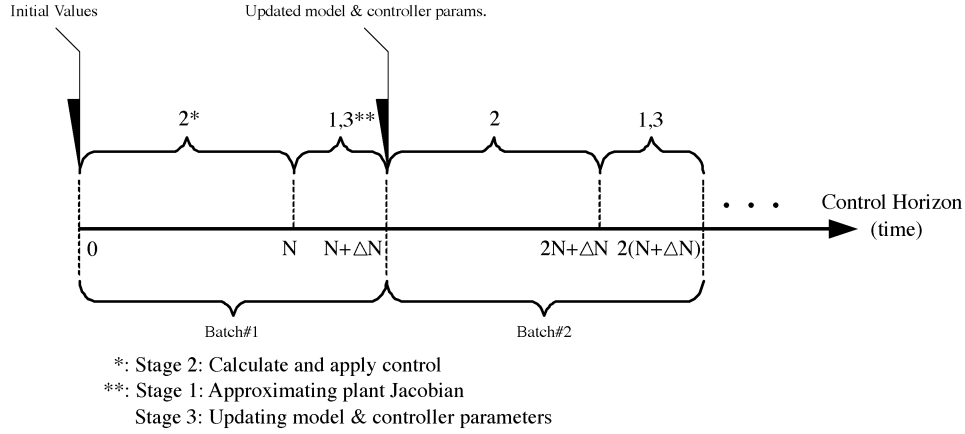
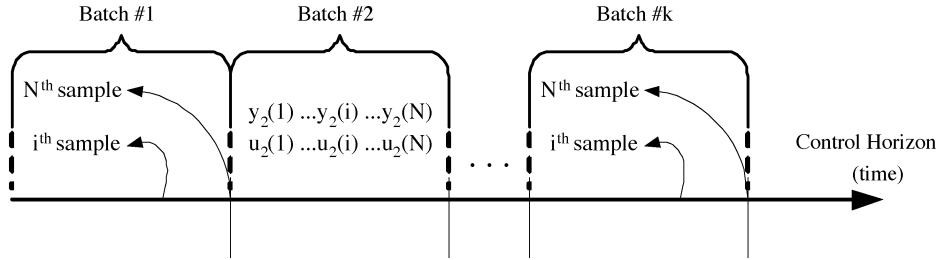


Fig. 1. ILC-based control design scheme.

Fig. 2. Control horizon is divided to k batches with N samples within each.

The assumption on boundness of non-Gaussian noise is made because most practical engineering systems are subjected to noises that are both bounded and of non-Gaussian nature. Examples are seen in most robust controller design where it has been generally assumed that the external disturbances are bounded with a known upper bound. The noise of the experimental system in this paper is upper bounded by the maximum output flow rate. The key issue here is to obtain a good estimate of interval $[a, b]$ from system (3). Indeed, this can be achieved once the model dynamics $f_p(\dots)$ is available. For unknown $f_p(\dots)$, an initial training of $f_p(\dots)$ under zero noise should be carried out using a neural network. In the case that the trained $f_p(\dots)$ is monotonic with respect to the noise term, such a boundness of the noise can be directly estimated from the measured inputs and outputs via the inverse of the initially trained $f_p(\dots)$ in (3). If the initial model of $f_p(\dots)$ cannot be obtained because the noise cannot be set to zero, then one has to use the knowledge of practical systems to locate the noise of the system and directly estimate interval $[a, b]$.

Although this assumption might seem strong, it does not really affect the pdf modeling. This assumption is made only to obtain strict bounded-input–bounded-output (BIBO) stability and it can be relaxed if the performance can be achieved with a probability sense.

To make the design procedure easier to follow, define

$$\rho_k(i) = \{y_k(i-1), \dots, y_k(i-n_a), u_k(i-n_d-1), \dots, u_k(i-n_b-n_d+1)\}^T. \quad (4)$$

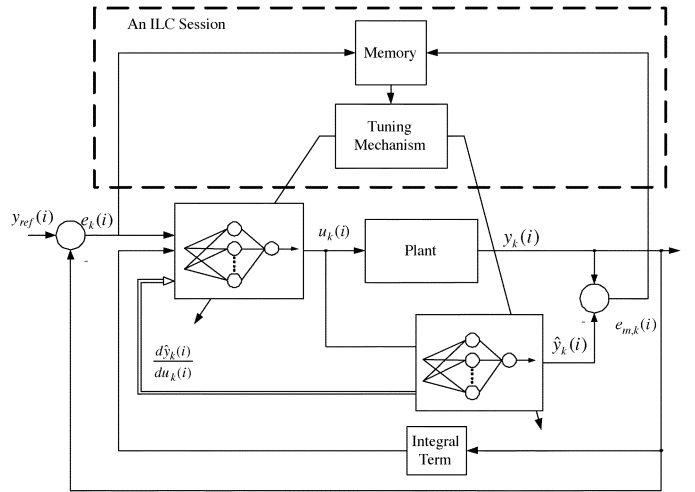


Fig. 3. Proposed scheme of ILC-based minimum entropy control.

Denoting the set point as $r_k(i)$, the output tracking error can be expressed as

$$e_k(i) = r_k(i) - y_k(i) = r_k(i) - f_p(\rho_k(i), u_k(i-n_d), \omega_k) \quad (5)$$

which is also a non-Gaussian random process. The aim of the controller design here is to determine a control signal u_k so that the entropy of the tracking error is decreased batch by batch. As discussed in the introduction section, this can realize a decrease of randomness for the closed-loop system. Fig. 3 illustrates the general scheme of the method proposed.

To design such a controller, the objective function to be minimized is expressed as follows:

$$J(\Theta) = H_k(r_k(i) - f_p(\rho_k(i), f_c(\Theta, r, y), \omega_k(i))). \quad (6)$$

In (6), $H_k(\cdot)$ is the entropy operation defined in (1). The controller f_c is a three-layer neural network and $\Theta = [\theta_1, \dots, \theta_P]^T$ is the parameter vector containing p weight and bias elements in f_c . These parameters are trained by an ILC tuning mechanism, which takes place between any two adjacent batches. For the entropy optimization purpose, the following Kernel density estimation method is used to estimate the tracking error pdf [11] within each batch using the sampled tracking error data:

$$\gamma_k(e) \approx \hat{\gamma}_k(e) = \frac{1}{N} \sum_{i=1}^N K_\sigma(e - e_i). \quad (7)$$

Using $\{e_1, \dots, e_N\}$ within each batch, the above pdf can be obtained where $K_\sigma(\cdot)$ is a real, symmetrical Kernel function with the specifications stated in [11] and $\sigma > 0$ can be considered as a bandwidth. The chosen Kernel function in this paper is expressed below

$$K_\sigma(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right). \quad (8)$$

The choice of the Kernel function is actually dependent on the level of smoothness the designer expects from the pdf estimation. Therefore, the discrete ILC-based controller tuning objective function can be rewritten as follows:

$$J(\Theta) = H_k(e) = \frac{1}{1-\alpha} \log(V_{R\alpha}(e_k)) \quad (9)$$

where $V_{R\alpha}(e)$ is often called the information potential (IP), which is expressed as

$$V_{R\alpha}(e) = \frac{1}{N^\alpha} \sum_{i=1}^N \left[\sum_{j=1}^N K_\sigma(e_k(i) - e_k(j)) \right]^\alpha. \quad (10)$$

The solution to this problem is studied in Section III. Since the selected kernel function satisfies $K(0) = 1$, the minimization of J would also mean that the tracking error magnitude is minimized. This ensures that the system output should be made as close as possible to the set point.

III. ILC-BASED SOLUTION TO THE PROBLEM

ILC was first introduced for systems with similar operation during certain time intervals, i.e., batches [22]. Examples are line manufacturing robots or chemical batch process controls [23]. In ILC, the control signal in the k th batch is based on the

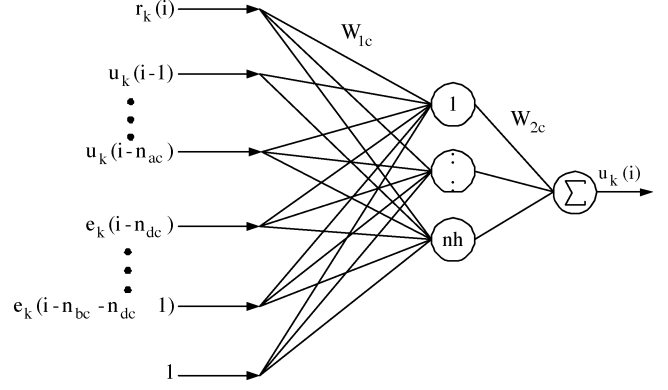


Fig. 4. Dynamic neural network controller structure.

control signal in the $(k-1)$ th batch and a correcting function, which is basically dependent on the closed-loop system performance in the $(k-1)$ th batch. A typical P-type ILC law can be written as follows:

$$u_k(i) = u_{k-1}(i) + \lambda J_{k-1}(i) \quad (11)$$

where $J_{k-1}(i)$ represents a function expressing the tracking performance index, e.g., difference between the desired output and the system output, and λ is the learning rate, chosen so that the iterative control law is convergent [24], [25]. In this paper, a modified ILC framework based on the three-layer dynamic feed-forward neural network is considered. In this regards, the parameters of the neural network model and controller are tuned based on ILC scheme between the batches.

A. Neural Network Controller Design

A three-layer dynamic neural network is considered as the adaptive controller with structure shown in Fig. 4.

In Fig. 4, n_{ac} , n_{bc} , and n_{dc} stand for dynamic structural orders of the controller and nh shows the number of hidden units. Input-to-hidden and hidden-to-output layer weight matrices are denoted by W_{1c} and W_{2c} . Thus, the controller parameter vector Θ is composed of W_{1c} and W_{2c} together with the biases of each layer.

The solution to tune the control parameters between adjacent batches involves a nonlinear programming algorithm where the objective function is not necessarily convex. Thus, the following pseudo-D-type ILC parameter updating law can only guarantee the local optimality:

$$\Theta_{k+1} = \Theta_k - \lambda_c^{\text{ILC}} \frac{\partial H_k(\Theta)}{\partial \Theta_k} \quad (12)$$

where λ_c^{ILC} is the ILC learning rate and Θ_k is the controller parameter vector within the k th batch. Using the well-known chain rule, the gradient part of (12) can be further expressed as follows:

$$\frac{\partial H_k(e(i))}{\partial \theta_k(p)} = \frac{\partial H_k(e(i))}{\partial V_k(e(i))} \times \frac{\partial V_k(e(i))}{\partial e_k(i)} \times \frac{\partial e_k(i)}{\partial \theta_k(p)}. \quad (13)$$

This leads to the following calculation:

$$\begin{aligned} \frac{\partial H_k(e)}{\partial \theta_k(p)} &= \frac{\alpha}{(1-\alpha)} \frac{1}{\sum_{i=1}^N \left[\sum_{j=1}^N K_\sigma(e_k(i) - e_k(j)) \right]^\alpha} \\ &\times \sum_{i=1}^N \left\{ \left[\sum_{j=1}^N K_\sigma(e_k(i) - e_k(j)) \right]^{\alpha-1} \right. \\ &\times \left[\sum_{j=1}^N \dot{K}_\sigma(e_k(i) - e_k(j)) \right. \\ &\times \left. \left. \left(\frac{\partial e_k(i)}{\partial \theta_k(p)} - \frac{\partial e_k(j)}{\partial \theta_k(p)} \right) \right] \right\} \quad (14) \end{aligned}$$

where $p = 1, 2, \dots, P$ and V stands for the IP.

The derivative of the tracking error with respect to the parameters can be written as

$$\begin{aligned} d_k(i) &= \frac{\partial e_k(i)}{\partial \theta_k(p)} = -\frac{\partial y_k(i)}{\partial \theta_k(p)} = -\frac{\partial y_k(i)}{\partial u_k(i)} \times \frac{\partial u_k(i)}{\partial \theta_k(p)} \\ &= -\Phi_p(i) \times \Phi_c(i) \quad (15) \end{aligned}$$

where $\Phi_p(i)$ and $\Phi_c(i)$ are the plant Jacobian and the controller gradient with respect to its parameters, respectively. However, the plant Jacobian is not available because the plant is unknown. A solution to this is to substitute the plant Jacobian with its approximation $\hat{\Phi}_p$ made by an identification procedure that uses another neural network to model the unknown plant. This means that

$$d_k(i) \simeq \hat{d}_k(i) = -\hat{\Phi}_p(i) \times \Phi_c(i) \quad (16)$$

where $\hat{\Phi}_p(i)$ is the estimated plant Jacobian using the neural network model of the plant. Thus, the ILC parameters tuning law can be summarized as (12) together with (13)–(16). In Section III-B, the neural-network-based identification of the plant will be described.

B. Nonlinear Plant Identification

Similar to the controller neural network, a neural network is employed to model the plant, leading to an approximation of the plant Jacobian, i.e., $\hat{\Phi}_p$. Denoting n_{am} , n_{bm} , and n_{dm} as dynamic structural orders of the model and h_m as the number of hidden units, the neural network model is shown in Fig. 5 with the following model output:

$$\hat{y}_k(i) = f_m(\phi) \quad (17)$$

where the regression vector ϕ is expressed as

$$\phi = \{y(i-1), \dots, y(i-n_{am}); u(i-n_{dm}), \dots, u(i-n_{bm}-n_{dm}+1)\}^T. \quad (18)$$

Then the modeling error can be written as

$$e_{m,k}(i) = \hat{y}_k(i) - y_k(i) = f_m(\phi) - y_k(i) \quad (19)$$

which is also a non-Gaussian random process, where i indicates the time samples within each batch. Since $e_{m,k}(i)$

is non-Gaussian, the use of the mean squared error for the modeling phase is not generally suited for the same reasons per discussed in the introduction section. As such, similar to the control tuning, the modeling should also be performed by the minimization of the modeling error entropy. This leads to the following objective function for the plant modeling phase:

$$J_m(\Theta_m) = H_k(e_m) = \frac{1}{1-\alpha} \log(V_{m,R\alpha}(e_{m,k})) \quad (20)$$

where the pdfs of the modeling error within each batches are calculated in the same way as in the controller design phase. As such, the same parameter update rule as the controller, i.e., pseudo-D-type ILC, is applied to the model parameters as follows:

$$\theta_{m,k+1}(p) = \theta_{m,k}(p) - \lambda_m^{\text{ILC}} \frac{\partial H_k(e_m)}{\partial \theta_{m,k}(p)} \quad (21)$$

together with

$$\begin{aligned} \frac{\partial H_k(e_m)}{\partial \theta_{m,k}(p)} &= \frac{\alpha}{(1-\alpha)} \frac{1}{\sum_{i=1}^N \left[\sum_{j=1}^N K_\sigma(e_m(i) - e_m(j)) \right]^\alpha} \\ &\times \sum_{i=1}^N \left\{ \left[\sum_{j=1}^N K_\sigma(e_m(i) - e_m(j)) \right]^{\alpha-1} \right. \\ &\times \left[\sum_{j=1}^N \dot{K}_\sigma(e_m(i) - e_m(j)) \right. \\ &\times \left. \left. \left(\frac{\partial \hat{y}(i)}{\partial \theta_m(p)} - \frac{\partial \hat{y}(j)}{\partial \theta_m(p)} \right) \right] \right\} \quad (22) \end{aligned}$$

where $p = 1, 2, \dots, P_m$ and λ_m^{ILC} is the modeling learning rate. Again, since the selected kernel function $K(0) = 1$, the minimization of J_m would also mean that the modeling error magnitude is also minimized. This ensures that the model output is made as close as possible to the system output. Therefore, the design procedure is summarized as follows.

- 1) Use a sequence of measured output and control signals from the $(k-1)$ th batch to form the modeling error vector as stated in (19).
- 2) Update Θ_m applying (20)–(22) and extract W_{1m} and W_{2m} .
- 3) Use W_{1m} and W_{2m} to calculate $\hat{\Phi}_p$.
- 4) Update Θ using (12)–(16) and extract W_{1c} and W_{2c} .
- 5) Calculate and apply the resulting control signal to the plant and return to stage 1).

Since the ILC algorithms involve the learning parameters, it is important to make sure that the learning rates are properly chosen to guarantee the convergence of the above proposed algorithm.

IV. ILC CONVERGENCE ANALYSIS

For simplicity, we only discuss the sufficient convergence conditions of the controller ILC algorithm because similar conclusions can also be formulated for the neural network modeling algorithm for the plant. The key issue is that the learning rate

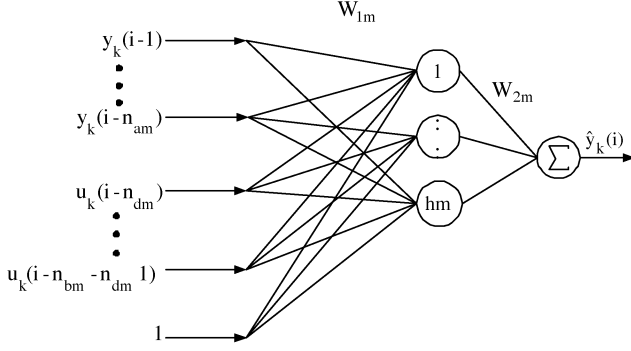


Fig. 5. Dynamic neural network model structure.

should cause the batch-by-batch decrease of (6). Therefore, the conditions are summarized in the following theorem.

Theorem 1: The entropy of closed-loop tracking error will be strictly decreasing with respect to batches (batch-by-batch decrement), if for all i, j , the chosen learning rate λ_c^{ILC} can justify the following nonlinear inequality:

$$\dot{K}_\sigma(e_k(i) - e_k(j)) \times (\hat{d}_k(i) - \hat{d}_k(j)) \lambda_c^{\text{ILC}} \frac{\partial H_{k-1}(e)}{\partial \theta_{k-1}} < 0. \quad (23)$$

Proof: Assuming that the current batch number is k , the convergence definition mentioned above would mean that the tracking error entropy is a decreasing function along with the batches k . This is equivalent to

$$H_k(e(i)) < H_{k-1}(e(i)). \quad (24)$$

Since $\alpha > 0$, then inequality (24) can be rewritten as

$$\log(V_{R\alpha}(e_{k-1}(i))) - \log(V_{R\alpha}(e_k(i))) < 0 \quad (25)$$

which means that

$$\log\left(\frac{V_{R\alpha}(e_{k-1}(i))}{V_{R\alpha}(e_k(i))}\right) < 0 \quad (26)$$

which is equivalent to the following conditions:

$$\frac{V_{R\alpha}(e_{k-1}(i))}{V_{R\alpha}(e_k(i))} < 1. \quad (27)$$

Since the IP is nonnegative, condition (26) would mean

$$\Delta V_{R\alpha,k} = V_{R\alpha}(e_k) - V_{R\alpha}(e_{k-1}) > 0. \quad (28)$$

This implies that the function $V_{R\alpha}$ should be an increasing function with respect to k . Such an implication will result in the following condition for ILC convergence:

$$\frac{\partial V_{R\alpha,k}}{\partial k} > 0. \quad (29)$$



Fig. 6. Overall view of byronic PCU test rig system.

Using the same principles as (14)–(16), it can be verified that condition (29) will result in the following inequality:

$$\sum_{i=1}^N \left\{ \left[\sum_{j=1}^N K_\sigma(e_k(i) - e_k(j)) \right]^{\alpha-1} \times \left[\sum_{j=1}^N \dot{K}_\sigma(e_k(i) - e_k(j)) (\hat{d}_k(i) - \hat{d}_k(j)) \Delta \theta_k \right] \right\} > 0 \quad (30)$$

where $\Delta \theta_k = \theta_k - \theta_{k-1}$.

As the term $\sum_{j=1}^N K_\sigma(e_k(i) - e_k(j))$ specifies a probability density estimation, it will be always nonnegative. Therefore, the convergence condition can be summarized to give

$$\sum_{i=1}^N \sum_{j=1}^N \left[\dot{K}_\sigma(e_k(i) - e_k(j)) (\hat{d}_k(i) - \hat{d}_k(j)) \Delta \theta_k \right] > 0. \quad (31)$$

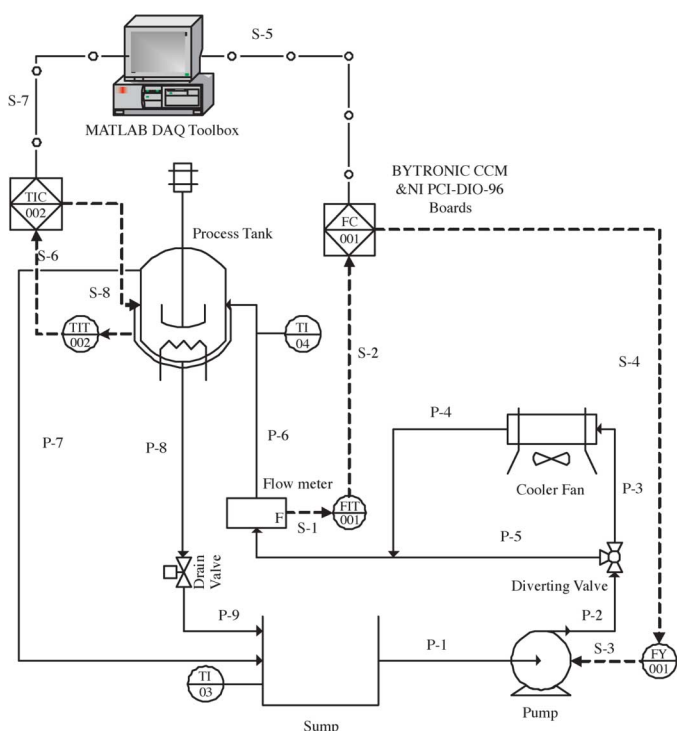
Replacing the term $\Delta \theta_k$ by its equivalent from (12), the convergence condition can be summarized to give (23).

This means that the iterative tuning rule (12) should be applied with the learning rate satisfying (23). ■

The example in the following section shows the effectiveness of the method proposed.

V. APPLICATION TO A PRACTICAL CONTROL SYSTEM

The proposed method has been successfully evaluated by a byronic laboratory-based process control unit (PCU). The PCU provides both hardware and software to implement different control algorithms based on microcomputer technology. However, some major changes were made on both original hardware and software, which will be introduced in upcoming sections. The general scheme of the test rig is shown in Fig. 6.



A. Process Description

The water is pumped out of the reservoir, i.e., sump, by a 12-V direct current (dc) centrifugal pump. The pump is able to create a variable flow rate from 0 to 2.0 l/min controlled by FC-001 computerized controller. Based on process requirements, the flow in P-2 pipeline can be diverted to either P-3 through the air-based cooling system or P-5 line. In both cases, the fluid passes through a six-blade impeller type flow meter FIT-001 on pipeline P-6 before it is directed to the main process tank. The process temperature inside the tank can be controlled by a 2.4-kW 220-V alternating current (ac) heating element. In order to have a homogenously heated product, a 12-V dc, 300-r/min dc motor has been foreseen on the top of the process tank to agitate the fluid inside. As such, the process tank can be referred to as a continuously stirred tank reactor (CSTR).

From thermodynamics, the flow variations affect the temperature, however process tank temperature does not have any effect on flow. As a result, the cooling fan can be regarded as a temperature compensation, whenever the flow affects the temperature. It should be noted that the sump temperature and flow temperature are measured and displayed on the PCU panel by TI-03 and

Loop	PV	MV
001	Process Flow Rate	Pump Speed
002	Tank Temperature	Heater Power

It must be noted that for the purpose of minimum entropy control, only the flow control loop, i.e., loop 001 has been considered and all the upcoming simulation results refer to the corresponding loop as mentioned.

All the temperature and flow signals are first converted to 0–5-V dc signals and then to 8-b digital signals where they are sent through the input/output (I/O) bus to the PC-based control system. For instance, the flow meter output is a 0–570-Hz pulse proportional to the flow rate. This signal is put through a frequency to voltage and then an analog-to-digital (A/D) converter to produce the 8-b digital signal. All the required signal conditioning and conversions are conducted in a Bytronic subsystem named as computer control module (CCM). The original I/O interface PC card is a SI-8255 IBM PC/XT/AT compatible card, which restricts the programming and control coding to *QBasic*. For ability to work under other programming environments such as MATLAB, the existing SI-8255 module was replaced by a National Instruments (NI) PCI-DIO-96, 96 channel 5V TTL/CMOS digital I/O module. As such, the existing wiring system had to be modified. The modified wiring system is shown in Fig. 8 [28].

There are two 40-pin data ports named CN1 and CN2 located on CCM that are used to transfer the data from the plant to the PC card. As such, two multicore cables (1) and (2) were used to physically split CN port pins and connect them to UMK-SE screw-type terminal blocks. UMK-SE terminal blocks provide screw-type terminals at the input side and 50-pin ribbon cable connection at the output side. It should be noted that before connecting 40-pin ports to UMK-SE terminal blocks, each pin on CN ports had to be tested in order to identify the pins corresponding to plant-side quantities. A special 2-to-1 ribbon cable CB-100 is then used to connect the UMK-SE to the PCI-DIO-96 I/O module. After identification of each port/channel of the I/O module, MATLAB Data Acquisition (DAQ) toolbox is used to communicate with the plant digitally [29].

After these modifications, several initial tests and analysis such as step response and frequency analysis were conducted before being able to implement minimum entropy algorithm. Having the system sampling time and cutoff frequency known, the main ILC-based minimum entropy mechanism was launched.

In order to evaluate the ILC-based minimum entropy, control loop 001 in Table I is considered as system under control and step input experiment is applied. The sampling time of the

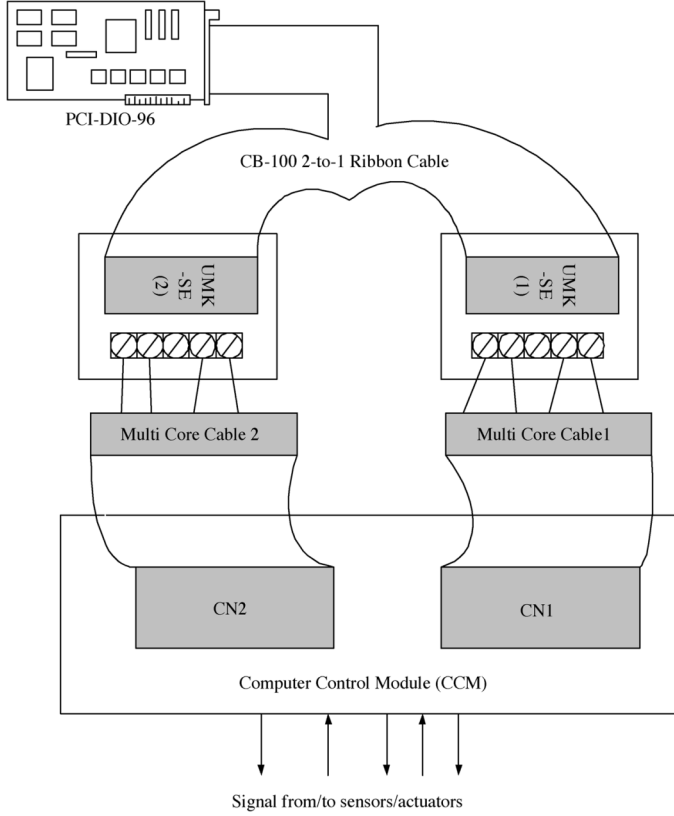


Fig. 8. Modified connection diagram of the test rig control system.

TABLE II
DYNAMIC ORDERS OF MODEL AND CONTROLLER

Model	Controller
$n_{am} = 6$	$n_{ac} = 5$
$n_{bm} = 3$	$n_{bc} = 3$
$n_{dm} = 1$	$n_{dc} = 1$

system is set to 100 ms, i.e., $T_s = 100$ ms. In order to avoid applying excessive control effort to the pump, the control is limited to $0 \leq u_k(i) \leq 10$ V dc and the set point is $y_k(i) = 1.0$ l/min. All the initial values are set to zero except Θ_1 , which is considered random. The number of hidden neurons in model and controller are set to $h_m = 5$ and $n_h = 6$, respectively. In addition, the dynamic orders of model and controller neural networks are listed in Table II.

The hyperbolic tangent and linear activation functions form all hidden and output neurons. The system runs under 20 batches each composed of 50 time samples, i.e., 5 s. The Kernel function is considered as

$$K_{0.25}(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2 \times 0.25^2}\right).$$

Finally, third-order Renyi's entropy measure is chosen, i.e., $\alpha = 3$ and initial learning rates are set to

$$\lambda_m^{\text{ILC}} = -0.2, \lambda_c^{\text{ILC}} = 10.$$

The plant behavior and dynamical orders n_a , n_b , and n_d are unknown. In practice, the model NN is pretrained by minimum squared error criterion using offline data to give the ILC-based

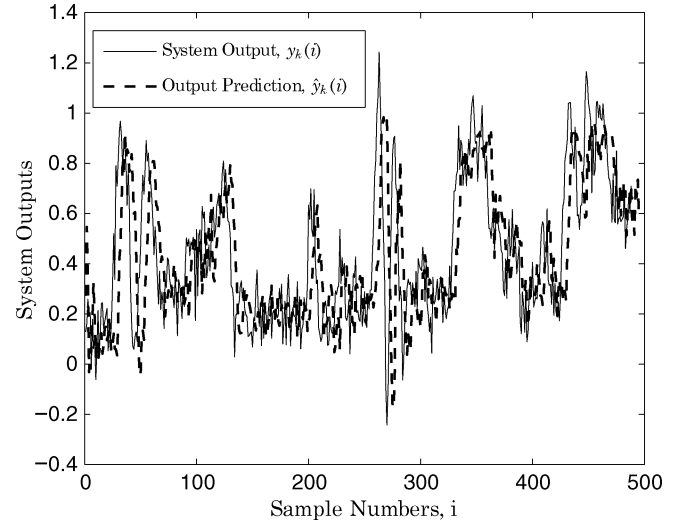


Fig. 9. One-step-ahead prediction capability of model.

online training routine “a good guess” of the initial network parameters. The training error associated with the pretrained NN model affects the robustness of parameter update laws as the first-order derivatives of the modeling error with respect to control signal appear in (13). Also, the training error affects the ILC convergence, as both first- and second-order derivatives of modeling error with respect to control signal are involved in (23). Therefore, the learning rate should be carefully chosen to satisfy (23) within the first batches of operation. Such effects were observed within the first batches of operation and then deteriorated as the ILC-based tuning as well as neural network model training were continued batch by batch. This confirms that proposed ILC-based solution can also reduce the effect of training error on control performance batch by batch. The model weights are initially chosen as random 5×10 and 1×6 matrices corresponding to W_{1m} and W_{2m} . From the second batch forwards, the tuning algorithms are applied. There are various methods to check the model validity among which the one-step-ahead prediction test and the error correlation methods are used in this paper. The one-step-ahead prediction check is performed by a fresh set of input data. The model is expected to track the output produced by the plant as close as possible. The result is shown in Fig. 9.

Also the tracking error autocorrelation and the cross correlation with the control signal can be evaluated to assess the validity of the model. With the residuals remaining within the bounds, model validity can be verified. Fig. 10 illustrates the results obtained from the tracking error autocorrelation (upper) and the cross correlation of the tracking error with the control signal (lower).

As mentioned, ILC trains the model parameters, i.e., model neural network weights, so that the entropy of modeling error is decreased along with the progress of the batches. Ideally, the entropy value in a typical batch k should be less than the entropy value in previous batch, i.e., $k - 1$. However, in practice, some variations can be observed due to unmodeled dynamics or slight parameter changes caused by mechanical or environmental effects. Although some small variations can be recorded,

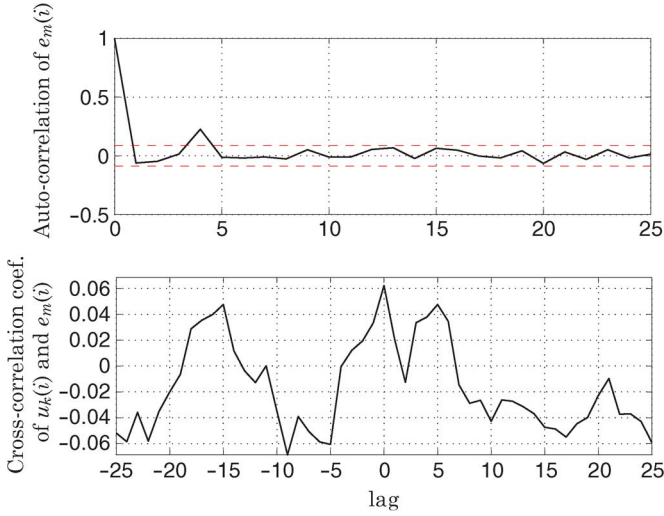
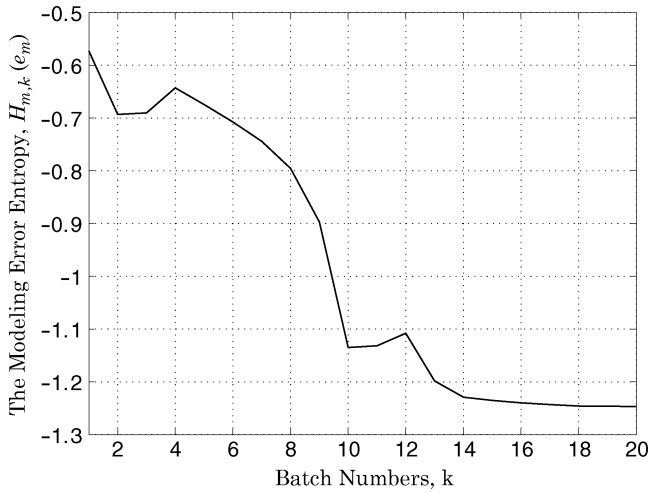


Fig. 10. Modeling error residual signals.

Fig. 11. Modeling error entropy $H_k(e_m(i))$ versus the ILC progress.

the adaptive tuning mechanism maintains the overall decreasing changes in entropy. This trend in the modeling error entropy is shown in Fig. 11.

Minimizing the error entropy measure should make the closed-loop modeling error pdf to approach a Gaussian alike and narrowly shaped pdf batch by batch, meaning that the randomness of the modeling mechanism within the closed-loop control system is minimized. This can be examined through a comparison between the modeling error pdf ($\gamma_{m,k}(e_m)$) at the first batch, i.e., $k = 1$ and the final batch when $k = 20$, Fig. 12 confirms such a change.

In addition to Fig. 12, the 3-D mesh of the modeling error pdf reconfirms that the model tracking error tends to take a Gaussian-alike shape with respect to the progress of the closed-loop operation. The results are illustrated in Fig. 13.

It must be noted that the model parameters are updated between any two adjacent batches but convergence does not take place until a certain number of batches have passed through. Although the convergence is guaranteed by a proper selection of learning rates described in Section IV, those learning rates excite different parameters by different rates, which may result

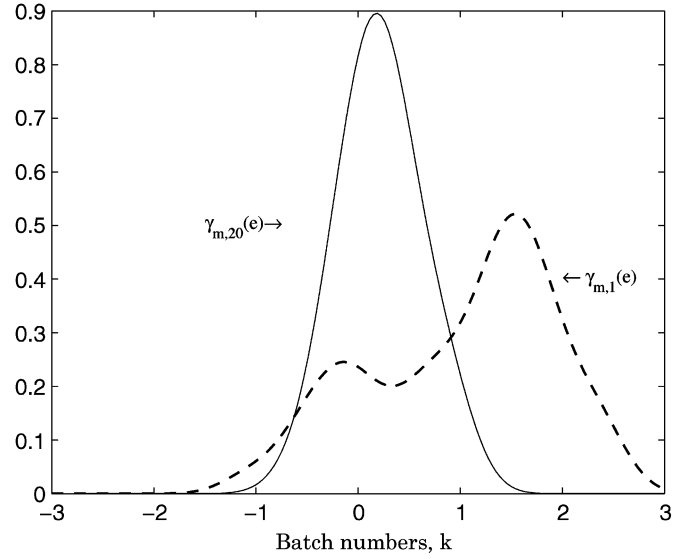


Fig. 12. Modeling error pdf in first and last batch of the ILC.

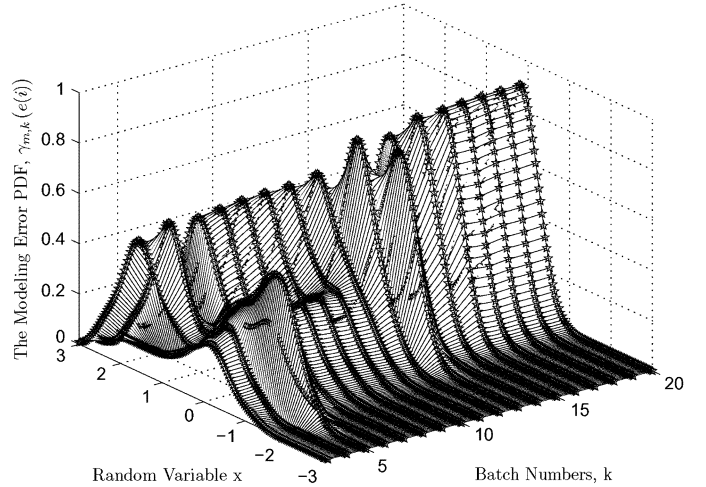


Fig. 13. The 3-D mesh of the modeling error pdf.

in parameter fluctuations in initial batches. However, as long as the learning rates satisfy the convergence conditions, this will not affect the ILC convergence or the closed-loop stability. With the mentioned structural dynamic order and the number of model hidden units mentioned before, the total number of the model parameters will be 56. Among those, the convergence, a selected number of parameters in each weight matrices W_{1m} and W_{2m} , namely, $W_{1m}(1,3)$, $W_{1m}(4,5)$, $W_{2m}(1,1)$, and $W_{2m}(1,2)$ along with the ILC closed-loop operation has been shown in Fig. 14.

As described before, the updated model parameters are used as a part of control design procedure immediately. Then, the ILC-based parameter tuning algorithm is applied to update the controller neural network parameters using the plant Jacobian data given by the model. After the controller parameters are updated, they will be kept constant to measure the control input within the upcoming batch. To evaluate the controller performance, one should note that the ILC algorithm is supposed to decrease the value of the tracking error entropy batch by batch.

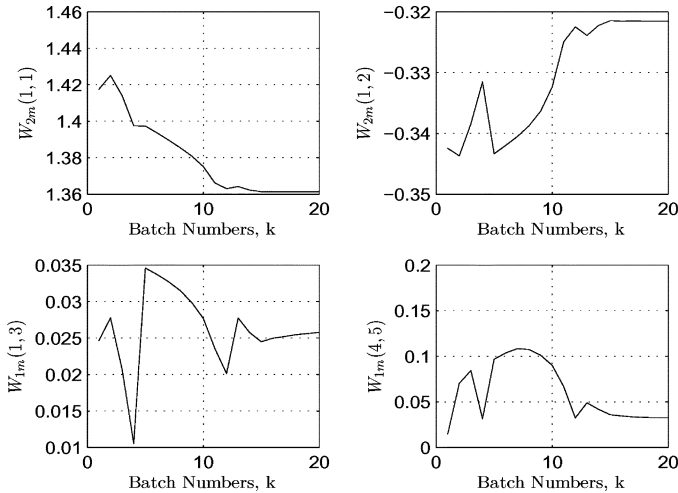


Fig. 14. Model neural network parameters converged through the ILC.

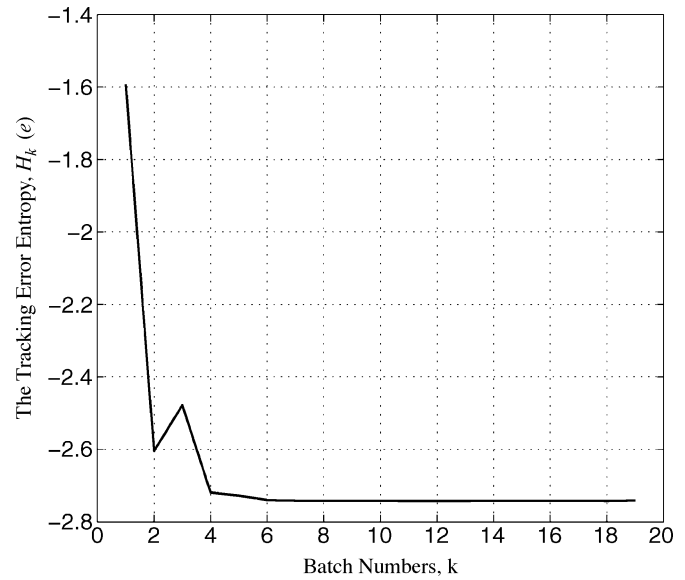


Fig. 16. Improvement of tracking error entropy through ILC.

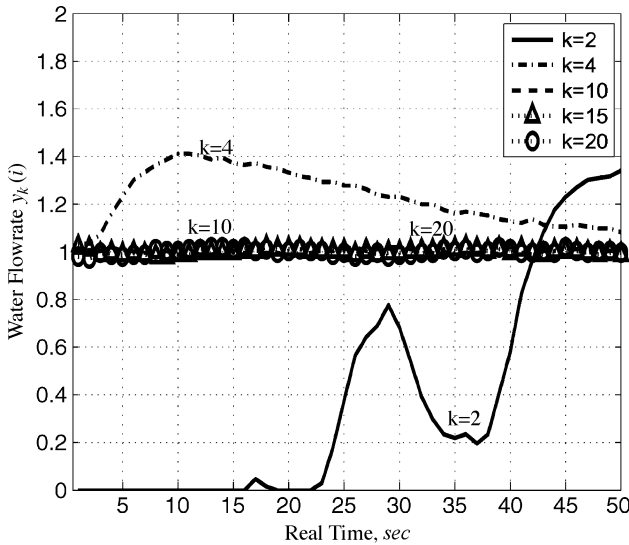


Fig. 15. Improvement of the set point tracking through ILC.

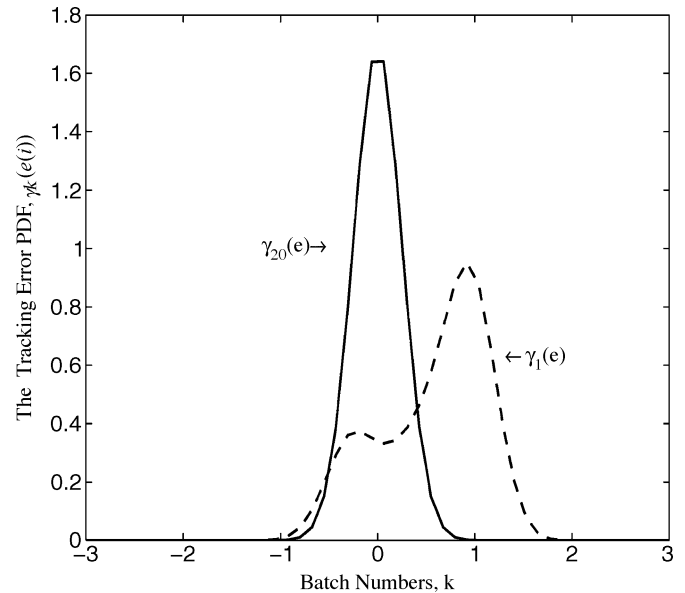
However, at the same time, the ILC should realize an improving set point tracking.

Fig. 15 illustrates how the tracking performance improves during selected batches, $k = 2, 4, 10, 15, 20$. As the batches of process control continue to run under the introduced the closed-loop ILC, a closer set point tracking can be achieved.

The ILC-based control signal thus calculated should result in the tracking error pdf to tend to a narrowly distributed shape along with the ILC closed-loop operation. The corresponding trend is shown as follows in Fig. 16.

Similar to the modeling error pdf, the pdf of the tracking error should dynamically change along with the batches and move towards a Gaussian-like pdf shape after 20 batches. Fig. 17 compares the probability density functions of the tracking error after the end of first batch ($k = 1$) and the final batch ($k = 20$).

Finally, the dynamic variations in 3-D mesh representation of the closed-loop tracking error distribution can be further studied to observe the dynamic transitional changes of the tracking error pdf along with the batches. Fig. 18 shows such a batch-by-

Fig. 17. Tracking error pdf changes from $k = 1$ to $k = 20$.

batch dynamic response in the tracking error pdf, ending with a Gaussian-like pdf shape for the tracking error.

VI. CONCLUSION

An ILC-based adaptive minimum entropy control method has been proposed for unknown nonlinear and non-Gaussian stochastic systems. The key idea is to divide the control horizon into a certain number of batches within which the control signal is applied to the plant, and between adjacent batches, the ILC-based minimum entropy tuning law is employed to update both model and controller parameters. Since the modeling and tracking error signals are non-Gaussian stochastic processes, the mean square of error minimization are not suited to characterize the randomness in the modeling and control phases. Thus, the goal of the optimization technique is to update model

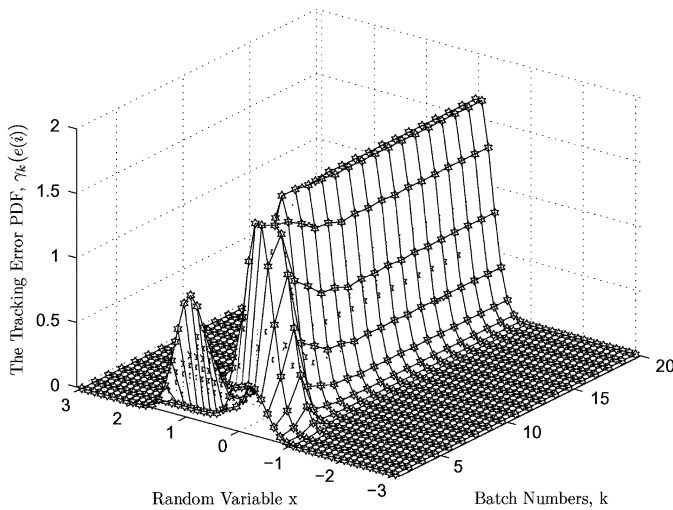


Fig. 18. The 3-D mesh of the tracking error through ILC.

and controller parameters (which are dynamic feedforward neural networks) in such a way that the entropy of modeling and tracing errors are minimized along with the progress of batches. In this regard, the neural network model plays a vital role because it provides the neural network controller with the approximated plant Jacobian. In fact, the proposed approach attempts to reduce the process plant randomness (i.e., make the probability density function of modeling and tracking errors as narrow as possible in a batch-by-batch manner). However, unlike in the previous papers by authors [20], [30], the proposed method does not rely on measurable pdf and can therefore be applied to any processes with nonlinearities and non-Gaussian noise distributions.

In addition, it provides a probabilistic view point to the entropy definition and quantifies the uncertainty definition, which is totally different from the information theoretic interpretation of entropy in H_∞ control methodologies. Furthermore, the proposed method makes considerable simplifications to the existing probabilistic minimum entropy control methods in terms of computational efficiency and ease of use in practical applications. First, thanks to Renyi's entropy definition, the tuning laws have been made simpler than those based on Shannon's entropy measure. Second, the mean value minimization is guaranteed without need to introduce an additional constraint to ILC performance function or conducting multiobjective optimization. This was possible with using a zero mean Kernel function to estimate the pdf and also adding an integral term to the neural network controller. Application of the proposed method to a functional laboratory test rig confirms the effectiveness of the proposed method.

REFERENCES

- [1] K. Astrom, *Introduction to Stochastic Control Theory*. New York: Academic, 1970.
- [2] V. Solo, "Stochastic adaptive control and martingale limit theory," *IEEE Trans. Autom. Control*, vol. 35, no. 1, pp. 66–71, Jan. 1990.
- [3] N. Filatov and H. Unbehauen, "Adaptive predictive control policy for nonlinear stochastic systems," *IEEE Trans. Autom. Control*, vol. 40, no. 11, pp. 1943–1949, Nov. 1995.

- [4] P. Shi, Y. Xia, G. Liu, and D. Rees, "On designing of sliding-mode control for stochastic jump systems," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 97–103, Jan. 2006.
- [5] H.-B. Ji and H.-S. Xi, "Adaptive output-feedback tracking of stochastic nonlinear systems," *IEEE Trans. Autom. Control*, vol. 51, no. 2, pp. 355–360, Feb. 2006.
- [6] H.-N. Wu and K.-Y. Cai, "Mode-independent robust stabilization for uncertain markovian jump nonlinear systems via fuzzy control," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 36, no. 3, pp. 509–519, Jun. 2006.
- [7] H. Wang, "Minimum entropy control of non-Gaussian dynamic stochastic systems," *IEEE Trans. Autom. Control*, vol. 47, no. 2, pp. 398–403, Feb. 2002.
- [8] H. Yue and H. Wang, "Minimum entropy control of closed-loop tracking errors for dynamic stochastic systems," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 118–122, Jan. 2003.
- [9] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–565, 1948.
- [10] N. Pal and S. Pal, "Entropy: A new definition and its applications," *IEEE Trans. Syst. Man Cybern.*, vol. 21, no. 5, pp. 1260–1270, Oct. 1991.
- [11] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1992.
- [12] H. Weidemann, "Entropy analysis of feedback control systems," in *Advances in Control Systems*. New York: Academic, 1969, pp. 225–255.
- [13] G. Saridis, "Entropy formulation of optimal and adaptive control," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 713–721, Aug. 1988.
- [14] X. Feng and K. Loparo, "Active probing for information in control systems with quantized state measurements: A minimum entropy approach," *IEEE Trans. Autom. Control*, vol. 42, no. 2, pp. 216–238, Feb. 1997.
- [15] I. Petersen, M. James, and P. Dupuis, "Minimax optimal control of stochastic uncertain systems with relative entropy constraints," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 398–412, Mar. 2000.
- [16] A. Wang and H. Wang, "Minimising entropy and mean tracking control for affine nonlinear and non-Gaussian dynamic stochastic systems," *Inst. Electr. Eng. Proc.—Control Theory Appl.*, vol. 151, no. 4, pp. 422–428, 2004.
- [17] H. Yue, J. Zhou, and H. Wang, "Minimum entropy of b-spline pdf systems with mean constraint," *Automatica*, vol. 42, no. 6, pp. 989–994, 2006.
- [18] D. Mustafa, K. Glover, and G. K. Paperback, *Minimum Entropy H_∞ Control*, ser. Lecture Notes in Control and Information science. Berlin, Germany: Springer-Verlag, 1990.
- [19] H. Wang, *Bounded Dynamic Stochastic Systems: Modeling and Control*. Secaucus, NJ: Springer-Verlag, 2000.
- [20] H. Wang, P. Afshar, and H. Yue, "ILC-based generalized PI control of output pdf of stochastic systems using LMI and RBF neural networks," in *Proc. IEEE Conf. Decision Control*, 2006, pp. 4163–4168.
- [21] H. Wang, H. Wang, J. Zhang, and H. Yue, "Periodic learning of b-spline models for output pdf control: Application to mwd control," in *Proc. Amer. Control Conf.*, J. Zhang, Ed., 2005, vol. 2, pp. 955–960.
- [22] A. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, pp. 123–140, 1984.
- [23] T. Al-Towaim, A. D. Barton, P. Lewin, E. Rogers, and D. H. Owens, "Iterative learning of control—2D control systems from theory to application," *Int. J. Control*, vol. 77, pp. 877–893, 2004.
- [24] Z. Bien and K. Huh, "Higher-order iterative learning control algorithm," *Inst. Electr. Eng. Proc. D—Control Theory Appl.*, vol. 136, no. 3, pp. 105–112, 1989.
- [25] D. H. Owens and J. Hätönen, "Iterative Learning Control—An optimization paradigm," *Annu. Rev. Control*, vol. 29, no. 1, pp. 57–70, 2005.
- [26] Instrumentation Society of America (ISA), *Instrumentation Symbols and Identification*, Research Triangle Park, NC, (R1992), 1984.
- [27] Bytronic, *Documentation for the Bytronic Process Control Unit* (IBM Version), U.K., 1998.
- [28] National Instruments (NI), *Digital I/O, 24 or 96 Lines, 5 V TTL/CMOS Data Sheet*, Austin, TX, 2007.
- [29] "Data Acquisition Toolbox User's Guide," Mathworks, Inc., 2006.
- [30] H. Wang and P. Afshar, "Radial basis function based iterative learning control for stochastic distribution systems," in *Proc. IEEE Int. Symp. Intell. Control*, 2006, pp. 100–105.



Puya Afshar (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, and the M.S. degrees from Tarbiat Modares (TMU), Tehran, Iran, in 2004 and the Ph.D. degree from Control Systems Centre, University of Manchester, Manchester, U.K., in 2007.

Since 2008, he has been a Research Associate in Advanced Control Systems under Prof. H. Wang's supervision. His research interests are stochastic distribution control, iterative learning control, multi-agent control systems, neural networks, and active fault tolerant control systems.



Hong Wang (M'95–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, Huazhong, China, in 1984 and 1987, respectively.

From 1988 to September 1992, he was a Research Fellow at Salford, Brunel, and Southampton Universities, U.K. He then joined UMIST in 2002, and has been a Professor in Process Control at the Control

Systems Centre, University of Manchester, Manchester, U.K., since 2002. He also holds a research visiting position with the Northeastern University (China), Huazhong University of Science and Technology (China), and Institute of Automation, Chinese Academy of Sciences. His research interests are stochastic distribution control, fault detection and diagnosis, nonlinear control, and data-based modeling for complex systems. He has published 200 papers and three books in these areas.

Prof. Wang was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and now serves as an editorial board member for the *Journal of Measurement and Control*, *Transactions of the Institute of Measurement and Control*, *IMechE Journal of Systems and Control Engineering*, *Automatica Sinica*, and *Journal of Control Theory and Applications*.



Tianyou Chai (M'90–SM'97–F'08) received the Ph.D. degree in control theory and engineering from Northeastern University, Shenyang, China, in 1985.

Since then he has been with the Center of Automation, Northeastern University, where he became a Professor in 1988 and a Chair Professor in 2004. He is the founder and Director of the Center of Automation, which became a National Engineering and Technology Research Center in 1997. His research interests include adaptive control, intelligent decoupling control, integrated plant control and

system, and the development of control technologies with applications to various industrial processes. He has made a number of important contributions in control technologies and applications. He has about 500 publications including 380 journal papers and 115 papers in international conferences, and three books. He has registered 12 invention patents among which two patents have been granted.

Prof. Chai was elected as a member of Chinese Academy of Engineering in 2003, and he was elected an academician of International Eurasian Academy of Sciences in 2007 and International Federation of Automatic Control (IFAC) Fellow in 2008. For his contributions, he has won three prestigious awards of National Science and Technology Progress. He received 2007 Industry Award for Excellence in Transitional Control Research, IEEE Multi-Conference on System and Control in Singapore. He also received Technological Science Progress Award from Ho Leung Ho Lee Foundation in 2002. He has been invited to give ten plenary speeches in international conferences of IFAC and IEEE.