Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

An ILC-based Minimum Entropy PI Controller for Unknown and Non-Gaussian Stochastic Systems

Puya Afshar, Hong Wang *

* *Control Systems Centre, The University of Manchester, PO BOX 88 M60 1QD UK (Tel: +44(0)161 306 4655; e-mail: hong.wang@ manchester.ac.uk).*

Abstract: In this paper a new adaptive control algorithm is presented for unknown nonlinear and non-Gaussian stochastic systems. The method combines the minimum entropy control with an Iterative Learning Control (ILC) framework, where the control horizon is divided into a number of time-domain intervals called *Batches*. Within each batch a PI controller is used to control the plant so as to achieve the required tracking performance, where a neural network is used to learn the dynamics of the unknown plant. Between any two adjacent batches, a D-type ILC law is applied to tune the PI control coefficients so that the tracking error entropy for the closed loop system is reduced batch by batch. The analysis on the ILC convergence is made and a set of demonstrable experiment results on a test rig are also provided to show the effectiveness of the obtained adaptive control algorithm.

Keywords: Iterative improvement, Stochastic distribution control, minimum entropy.

1. INTRODUCTION

Since most of practical control systems exhibit stochastic behaviour, research into stochastic control has been of a great importance during the past decades. Among various approaches, the minimum variance control Astrom (1970) is still one of the key methods in this research field. More recent works have also considered a variety of methods ranging from sliding mode control for jump stochastic systems Shi et al. (2006) to robust fuzzy control for uncertain Markovian stochastic systems Wu and Cai (2006). In most of the above methods, the closed loop tracking error has played a vital role for the control performance assessment, where most of the developed methods assume that the stochastic systems are subjected to Gaussian type noises. However, for systems subjected to non-symmetrically distributed random noises, the spread area of the noise distribution cannot be described precisely by using its variance. Therefore, entropy should be used as a measure of uncertainty in non-Gaussian stochastic systems. In fact, it is shown that for Gaussian systems, the minimum entropy control is equivalent to minimum variance control Wang (2002); Yue and Wang (2003).

Entropy was first introduced as a measure of uncertainty Shannon (1948). Later on, it was used as the average information content on randomness for a given Probability Density Function (PDF) Silverman (1992) which provided a generalized randomness measure by considering dispersion rather than mean or variance. Among the entropy definitions, the $\alpha-$ order Renyi's quadratic entropy (1) which has advantage of computational efficiency over others will be used throughout this paper.

$$H(y) = \frac{1}{1-\alpha}\log\left(\sum_i \gamma_i^{\alpha}(y))\right) \tag{1}$$

where $\alpha > 0$ and γ stands for the PDF of the concerned random variable y. The entropy measure has been applied to some of the control and estimation problems such as approximation, optimization and adaptive control associated with the uncertainty for controller design Saridis (1988) and the minimum entropy control for stochastic systems Wang (2002) and Yue et al. (2006). However, previous works have either considered the plant as Gaussian type or assumed known plants and noise distribution. In this paper, an ILC-based adaptive method is proposed for the non-Gaussian systems with unknown nonlinear dynamics. The key idea is to divide the control horizon into certain number of *batches* and control the plant *within a batch* and *between adjacent batches* as summarized as follows.

(1) Approximating the unknown plant Jacobian using a neural network model (Between batches);
(2) Calculating control signal (Within a batch);
(3) Updating the controller parameters using D-type ILC law and returning to (1)(Between batches).

This paper is organized as follows. In section 2, the the ILC-based minimum entropy control is introduced. Section 3 consists of the design details. The ILC convergence is analyzed in section 4, whilst the application of proposed method to a test rig is proposed in the section 5. Finally, concluding remarks are made in section 6.

2. PROBLEM FORMULATION

It is assumed that the unknown stochastic plant can be expressed as the following NARMAX model.

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

where f_p is the unknown nonlinear plant which is assumed continuous, bounded and first order differentiable with respect to all of its variables. As in Fig. 1, $y_k(i), u_k(i)$ are the i^{th} samples of the output and the control signal within the k^{th} batch, respectively. Also, n_a, n_b, n_d are the dynamical orders of the plant and ω_k is a bounded non-Gaussian random noise.

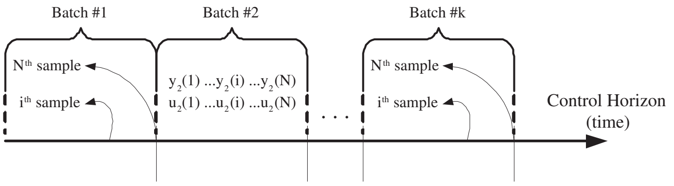

Fig. 1. Dividing the control horizon into batches

Denote the set point as $r_k(i)$, then the closed loop tracking error would be $e_k(i) = r_k(i) - y_k(i)$ which is generally a non-Gaussian random process. The objective is to determine $u_k(i)$ so that the entropy of the tracking error is decreased batch by batch. This means that the control input design aims at reducing the randomness of the tracking error along the progress of batches. Fig. 2 shows the general scheme of the method proposed.

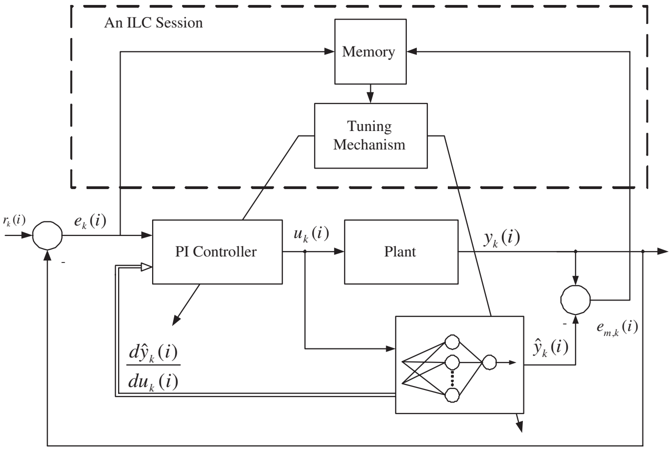

Fig. 2. Proposed ILC based minimum entropy control

The objective function to be minimized is expressed as

$$J = H_k(e) = \frac{1}{1-\alpha}\log(V_{R\alpha}(e_k)) \quad (3)$$

where $V_{R\alpha}(e)$ is the Information Potential (IP).The PDF required by IP is estimated as follows by the Kernel estimation method as in Silverman (1992).

$$\gamma_k(e) \approx \hat{\gamma}_k(e) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(e - e_i) \quad (4)$$

where K_α is a symmetrical Kernel satisfying $K(0) = 1$. Therefore, the IP can be denoted as

$$V_{R\alpha}(e) = \frac{1}{N^\alpha}\sum_{i=1}^{N}\big[\sum_{j=1}^{N} K_\sigma(e_k(i) - e_k(j))\big]^\alpha \quad (5)$$

The choice of the Kernel function depends on the level of required smoothness of the PDF estimation. Since the selected kernel satisfies $K(0) = 1$, the minimization of J would also mean that the tracking error magnitude is minimized. This ensures that the system output be as close to $r_k(i)$ as possible.

3. ILC-BASED SOLUTION TO THE PROBLEM

Since introduced in 1984 Arimoto et al. (1984), ILC has become one of the well received methods that have led to several industrial applications on line manufacturing systems and chemical batch processes Al-Towaim et al. (2004),Owens and Hätönen (2005), and Xu and Yan (2004). In this paper, the PI controller is tuned based on an ILC scheme to achieve minimum entropy performance.

3.1 Adaptive PI-Controller Design

A generalized PI controller with tuneable coefficients is considered as the adaptive controller as follows.

$$\begin{aligned} u_k(i) &= K_p(k)e_k(i) + K_i(k)\xi_k(i) \\ \xi_k(i) &= \xi_k(i-1) + T_s e_k(i-1) \end{aligned} \quad (6)$$

where T_s is the sampling time. The ILC-based solution to (3) involves a nonlinear programming where the objective function is not necessarily convex. Thus, the following parameter updating law can only guarantee the local optimality.

$$\begin{aligned} K_p(k+1) &= K_p(k) - \lambda_p \frac{\partial H}{\partial K_p}\bigg|_k \\ K_i(k+1) &= K_i(k) - \lambda_i \frac{\partial H}{\partial K_i}\bigg|_k \end{aligned} \quad (7)$$

In (7), $\lambda_p,\ \lambda_i$ are ILC learning rates and $K_p(k),\ K_i(k)$ are the PI controller coefficients within the k^{th} batch. Using the chain rule, the following training rules can be obtained.

$$\begin{aligned} \frac{\partial H_k(e)}{\partial K_p(k)} &= \frac{\alpha}{(1-\alpha)} \frac{1}{\sum_{i=1}^{N}\Big[\sum_{j=1}^{N} K_\sigma\big(e_k(i) - e_k(j)\big)\Big]^\alpha} \\ &\times \sum_{i=1}^{N}\Bigg\{\Big[\sum_{j=1}^{N} K_\sigma\big(e_k(i) - e_k(j)\big)\Big]^{\alpha-1} \\ &\times \Big[\sum_{j=1}^{N} \acute{K}_\sigma\big(e_k(i) - e_k(j)\big)\big(D_{p,k}(i) - D_{p,k}(j)\big)\Big]\Bigg\} \end{aligned} \quad (8)$$

where

$$D_{p,k}(i) = \frac{\partial e_k(i)}{\partial K_p(k)} = \frac{\partial e_k(i)}{\partial u_k(i)} \times \frac{\partial u_k(i)}{\partial K_p(k)} \simeq -\frac{\partial \hat{y}_k(i)}{\partial u_k(i)} e_k(i) \quad (9)$$

Calculations corresponding to the integral term are similarly performed to give

$$D_{i,k}(i) \simeq -\frac{\partial \hat{y}_k(i)}{\partial u_k(i)}\xi_k(i) \quad (10)$$

Thus the ILC parameters tuning law can be summarized as (7)-(9). The approximations made above are related to unknown plant dynamics yet to be determined. Therefore, the plant will be modeled to approximate Jacobian accordingly. This will be performed by an MLP neural network model described in the next section.

3.2 A Neural Network Based Nonlinear Plant Identification

Denoting n_{am}, n_{bm}, and n_{dm} as dynamic structural orders of the model and hm as the number of hidden units, the neural network model is shown in Fig.3 with the following model output.

$$\hat{y}_k(i) = f_m(y(i-1), \cdots, y(i-n_{am}); u(i-dm), \cdots, u(i-n_{bm}-n_{dm}+1), 1) = f_m(\phi_{m,k}) \tag{11}$$

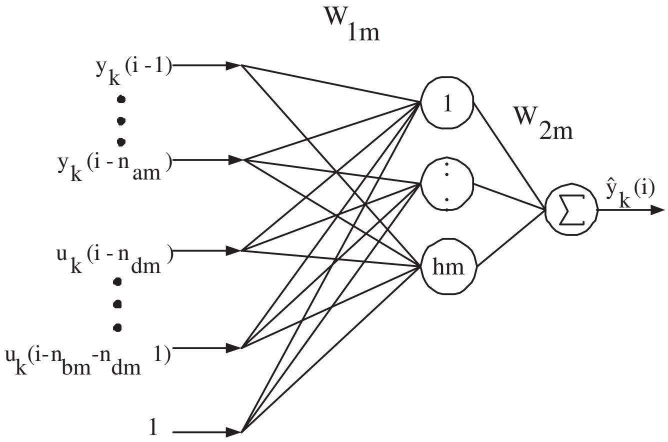

Fig. 3. The dynamic neural network model structure

The modeling error is denoted as $e_{m,k}(i) = \hat{y}_k(i) - y_k(i)$ which is also a non-Gaussian random process. As such, mean squared error criteria are not generally suited for the training of the neural network. Thus, modeling entropy minimization with the following objective function is used here.

$$J_m = H_k(e_m) = \frac{1}{1-\alpha}\log(V_{m,R\alpha}(e_{m,k})) \tag{12}$$

where the PDFs of the modeling error within each batches are calculated in the same way as those in the controller design phase. Then, the model parameter updating law can be formulated as follows.

$$\theta_m(k+1,p) = \theta_m(p,k) - \lambda_m \frac{\partial H_k(e_m)}{\partial \theta_m(p,k)} \tag{13}$$

together with

$$\frac{\partial H_k(e_m)}{\partial \theta_m(p,k)} = \frac{-\alpha}{(1-\alpha)} \frac{1}{\sum_{i=1}^{N}\left[\sum_{j=1}^{N} K_\sigma\big(e_m(i)-e_m(j)\big)\right]^{\alpha}} \times \sum_{i=1}^{N}\left\{\left[\sum_{j=1}^{N} K_\sigma\big(e_m(i)-e_m(j)\big)\right]^{\alpha-1} \left[\sum_{j=1}^{N} \acute{K}_\sigma\big(e_m(i)-e_m(j)\big)\Big(\frac{\partial \hat{y}(i)}{\partial \theta_m(p)} - \frac{\partial \hat{y}(j)}{\partial \theta_m(p)}\Big)\right]\right\} \tag{14}$$

where $p = 1, 2, \cdots, P_m$ and λ_m is the model learning rate. Choosing hyperbolic tangent activation functions for the neural network, it can be verified that the plant Jacobian can be calculated as follows

$$\frac{\partial \hat{y}_k(i)}{\partial u_k(i)} = W_{2m} \times W_{1m} \times \Big(1 - \tanh^2(W_{1m} \times \phi_{m,k})\Big) \tag{15}$$

Therefore, the design procedure is summarized as follows.

(1) Use $y_{k-1}(i)$ and $u_{k-1}(i)$ and form $e_{m,k-1}(i)$;
(2) Update $\theta_m(p,k)$ by (12)-(14) and extract W_{1m}, W_{2m};
(3) Use W_{1m}, W_{2m} and (15) to calculate Jacobian;
(4) Calculate $K_p(k)$ and $K_i(k)$ using (7)-(9) and (10);
(5) Calculate the control signal and apply it to the plant and return to (1).

An important issue in the above ILC-based design is to ensure that the chosen learning parameters guarantee the ILC convergence. This issue will be addressed in the next section.

4. ILC CONVERGENCE ANALYSIS

For simplicity, we only discuss the sufficient convergence conditions of the controller ILC algorithm. Similar conditions can also be formulated for the modeling phase. The key issue is that the convergence would mean a batch-wise decrease of (3) giving

$$H_{k+1}(e(i)) \le H_k(e(i)) \tag{16}$$

Since $\alpha > 0$ and IP is non-negative, then inequality (16) would mean

$$\Delta V_{R\alpha,k+1} = V_{R\alpha}(e_k) - V_{R\alpha}(e_{k+1}) \le 0 \tag{17}$$

Using (3), the following approximation can be made.

$$\begin{aligned}\Delta V_{R\alpha,k} &\approx \Delta\left[\frac{1}{N^\alpha}\sum_{i=1}^{N}\left[\sum_{j=1}^{N} K_\sigma\big(e_k(i)-e_k(j)\big)\right]^{\alpha}\right] \\ &= \frac{1}{N^\alpha}\sum_{i=1}^{N}\left\{\left[\sum_{j=1}^{N} K_\sigma\big(e_k(i)-e_k(j)\big)\right]^{\alpha-1} \left[\sum_{j=1}^{N} \acute{K}_\sigma\big(e_k(i)-e_k(j)\big)\Delta\big(e_k(i)-e_k(j)\big)\right]\right\}\end{aligned} \tag{18}$$

where

$$\Delta e_k(i) \approx \frac{\partial \hat{y}_k}{\partial u_k(i)} \times \left[e_k(i)\Delta K_p + \xi_k(i)\Delta K_i\right] \tag{19}$$

$$\begin{aligned}\Delta K_p &= K_p(k+1) - K_p(k) \\ \Delta K_i &= K_i(k+1) - K_i(k)\end{aligned} \tag{20}$$

As such, the for the ILC algorithm to converge, the learning parameters λ_p and λ_i must be chosen so that the (17)-(20) are satisfied.

5. PRACTICAL IMPLEMENTATION

The proposed method has been successfully implemented on a Process Control Unit (PCU) to be described in this section.

5.1 Process Description

The PCU is based on a fluid flow process, where fluid flow and/or temperature can be controlled. The PCU diagram is shown in Fig. 4, which is comprised of a sump, a pump, some solenoidal valves, a cooler fan and a process tank.

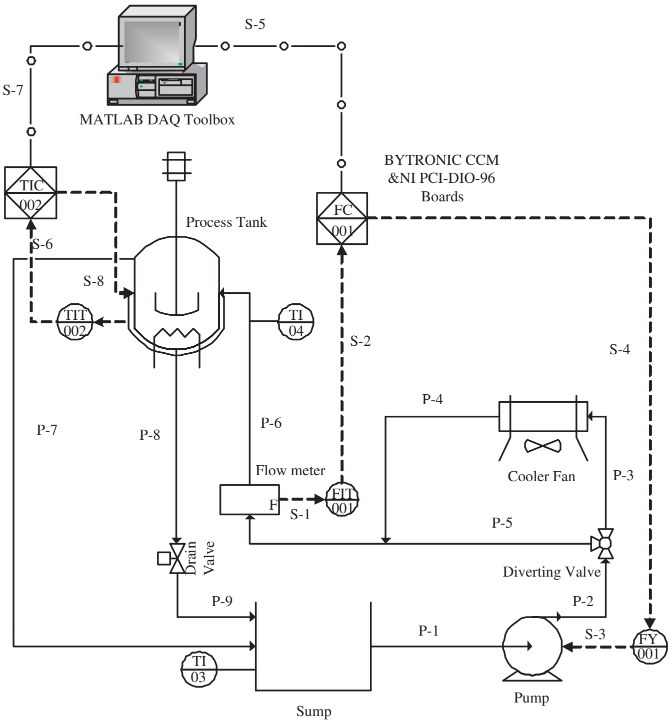

Fig. 4. PCU Piping and Instrumentation (P&I) Diagram

The water is pumped around by a 12 VDC centrifugal pump with a computer controlled variable flow rate ranging from 0 to 2.0 lpm. The flow in P-2 pipe can be diverted to either pathes P-3 through the cooling fan, or P-5. Then it passes through an impeller type flow meter FIT-001 on P-6 before it is directed to the process tank. The process values are the flow rate FIT-001 and the temperature TIT-002 and the manipulated variables are the pump speed FY-001 and the heater power TIC-002. However, only the flow control will be considered here.

5.2 Modifications to Original System

All physical process signals are firstly converted to 0-5 VDC and then 8-bit digital signals in a so called Computer Control Module (CCM). These signals are then transferred to PC I/O interface card $SI - 8255$ IBM PC/XT/AT compatible, this restricts the programming to *QBasic*. To develop the control algorithm in an MATLAB environment the existing I/O module was replaced by a National Instruments (NI) PCI-DIO-96, 96 channel digital I/O module and all relevant wiring designs were modified accordingly NI (2007).

5.3 Experimental Results

The T_s is set to 100 $msec$. The system runs under 60 batches for each $N = 50$. Also, $\alpha = 3$ and the K_σ are

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

Furthermore, the set point is supposed to change after 30 batches from $r = 1.3$ lpm to $r = 0.6$ lpm. Also a disturbance is applied at $k = 20$ and $k = 40$. Such disturbance is realized by opening the pump bypass valve, recycling a portion of water to be sump. The model NN is selected as $n_{am} = 6$, $n_{bm} = 3$, $n_{dm} = 1$ with $hm = 5$. In addition, $K_p(1) = 2.0$, $K_i(1) = 16$ and $\lambda_p^{ILC} = \lambda_i^{ILC} = 1$.

As mentioned, ILC trains the weights of the neural network model so that the entropy of modeling error is decreased along with the progress of the batches. This trend is shown in Fig. 5.

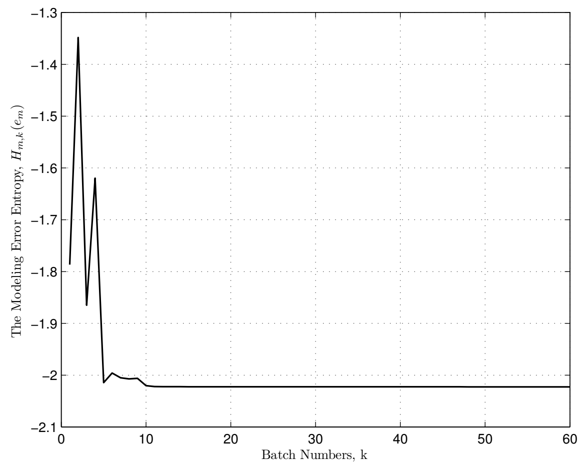

Fig. 5. $H_k(e_m(i))$ changes along with the batches

Minimizing the error entropy measure should make the PDF of the closed loop modeling error to approach a Gaussian alike and narrowly shaped PDF batch-by-batch. This means that the randomness of the modeling error 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 modelling error PDF at the final batch when $k = 60$ as shown in Fig.6, which confirms such a change in the modeling error PDF shape.

The updated model parameters are used to provide the controller with the approximated plant Jacobian as described before. In addition, the closed loop tracking performance should lead to a batch-wise improvement at the same time. Fig. 7 illustrates the controlled flow rate during the first 30 batches where the set point is $r = 1.3$ lpm. The figure confirms the improved trend in the closed-loop tracking performance.

The bypass valve of the pump is opened during the 20^{th} batch, resulting in an output drift within the 21^{st} batch. However, the tuning algorithm starts to minimize the tracking error, or in other words, to reject the disturbance.

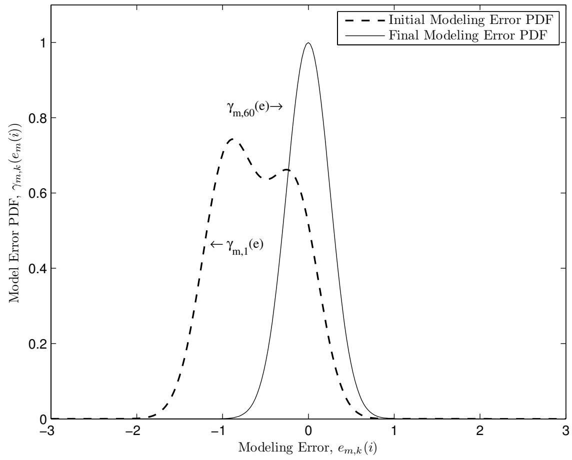

Fig. 6. The modeling error PDF $\gamma_{m,k}(e_m)$ from $k = 1$ to $k = 60$

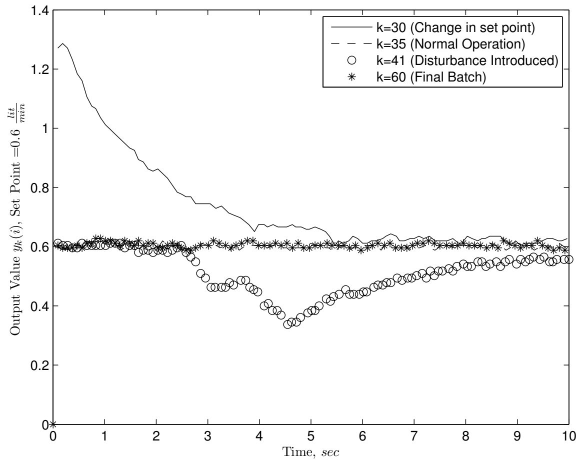

Fig. 8. The closed loop set point tracking through ILC for $r = 0.6$ lpm

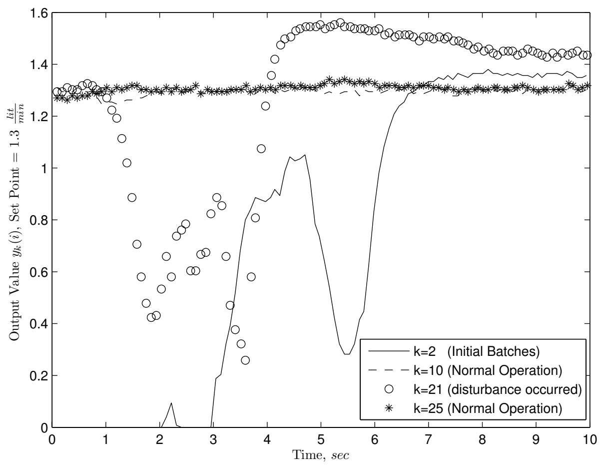

Fig. 7. The closed loop set point tracking through ILC for $r = 1.3$ lpm

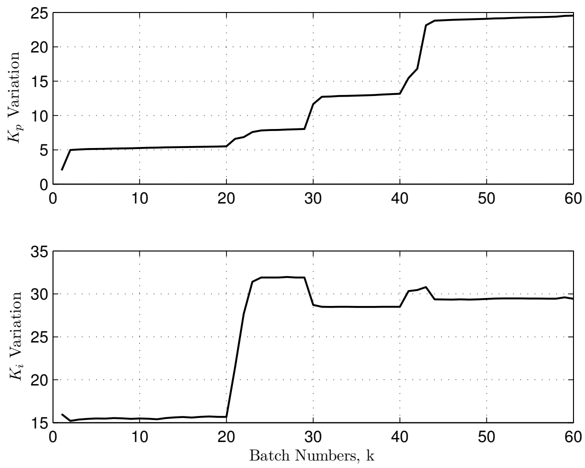

Fig. 9. Variations in the PI coefficients-(upper:K_p, lower:K_i)

At shown in Fig.8, the set point is changed to 0.6 lpm at 30^{th} batch. The set point tracking performance is achieved after 5 seconds. However, the pump bypass valve is opened again at beginning of the 40^{th} batch. The disturbance is successfully rejected by the adaptive system after 6 seconds into the 41^{st} batch.

It would be interesting to study PI controller coefficients along with the batches as illustrated in Fig. 9. The tuning algorithm tunes controller to $K_p(20) = 6$ and $K_i(20) = 15.5$ in order to minimize the entropy. Opening the pump bypass valve at $k = 21$ will decrease the volume of the water flowing through the $P - 5$ pipe, thus increasing the tracking error and making slight increase to K_P. Meanwhile, K_i increases more to eliminate the increase in the tracking error mean value. Then the set point is changed to $r = 0.6$ lpm at $k = 30^{th}$. The same disturbance, yet lasting for a short period of time, occurs at the 40^{th} batch and the tuning algorithm acts to maintain the desired performance.

The tuning algorithm should lead to minimization of the closed loop tracking error entropy along with the progress of batches. Fig.10 suggests that in spite of disturbances and set point changes, the entropy minimization has been effectively achieved.

The control tuning algorithm should result in a narrowly-shaped PDF of the tracking error. Fig. 11 suggests that in spite of the disturbances and changes in the operating point, $\gamma_k(e_k(i))$ will be shaped as a Gaussian-alike distribution at the end of the final batch.

Finally, Fig. 12 illustrates the dynamic variations in a 3D mesh representation of the closed loop tracking error distribution. This figure confirm that such a batch-by-batch dynamic change in the tracking error PDF has led to a Gaussian-alike PDF shape in the end.

6. CONCLUSIONS

An ILC based adaptive minimum entropy control method has been proposed for unknown nonlinear and non-

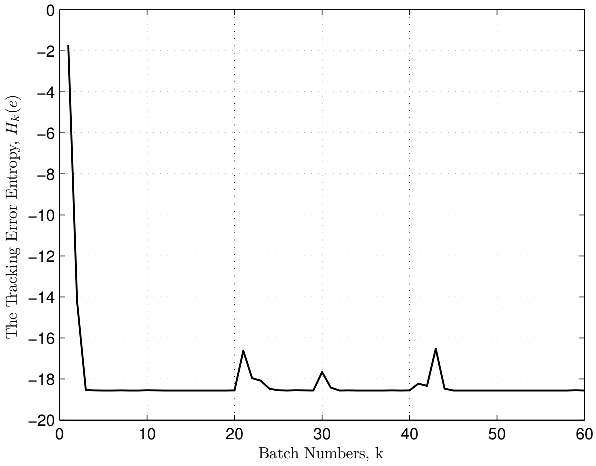

Fig. 10. Tracking error entropy

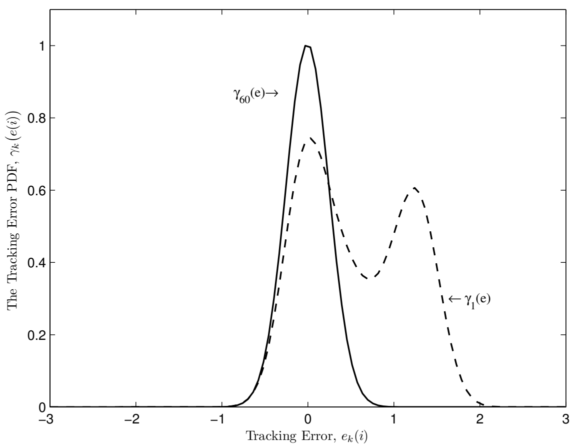

Fig. 11. The tracking error PDF from $k = 1$ to $k = 60$

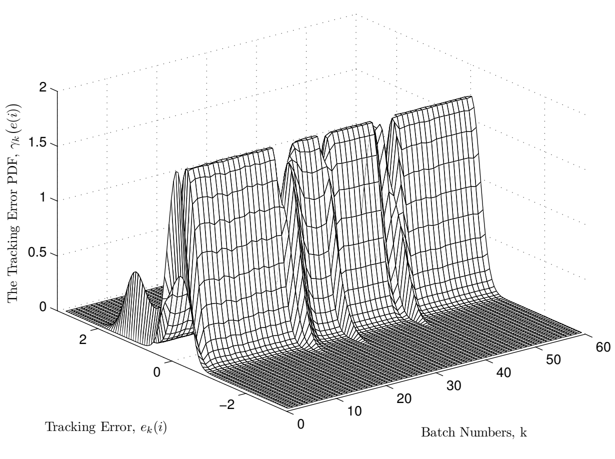

Fig. 12. The 3D mesh of the tracking error through ILC

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 plant model and controller parameters. Since the modeling and tracking error signals are non-Gaussian, 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 and controller parameters in such a way that the entropy of modeling and tracking errors are minimized along with the progress of batches. In this regard, the neural network model plays a vital role in terms of providing the controller with the plant Jacobian. The proposed algorithm has bee applied to a test rig where experimental results have confirmed the effectiveness of the proposed method.

REFERENCES

T. Al-Towaim, A. D. Barton, P.L. Lewin, E. Rogers, and D. H. Owens. Iterative Learning of Control - 2D Control systems from Theory to Application. *International Journal of Control*, 77:877–893, 2004.

A. Arimoto, S. Kawamura, and F.Miyazaki. Bettering Operation of Robots by Learning. *Journal of Robotic Systems*, 1:123–140, 1984.

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

NI. *Digital I/O, 24 or 96 Lines, 5 V TTL/CMOS Data Sheet.* National Instruments, 2007.

D. H. Owens and J. Hätönen. Iterative Learning Control- An optimization paradigm. *Annual Reviews in Control*, 29(1):57–70, 2005.

G.N. Saridis. Entropy formulation of optimal and adaptive control. *Automatic Control, IEEE Transactions on*, 33 (8):713–721, 1988. ISSN 0018-9286.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–565, July,October 1948.

P. Shi, Y. Xia, G.P. Liu, and D. Rees. On designing of sliding-mode control for stochastic jump systems. *Automatic Control, IEEE Transactions on*, 51(1):97–103, 2006. ISSN 0018-9286.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall/CR, 1992.

H. Wang. Minimum entropy control of non-gaussian dynamic stochastic systems. *Automatic Control, IEEE Transactions on*, 47(2):398–403, 2002. ISSN 0018-9286.

H. Wu and K. Cai. Mode-independent robust stabilization for uncertain markovian jump nonlinear systems via fuzzy control. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 36(3):509–519, 2006. ISSN 1083-4419.

J. X. Xu and R. Yan. Iterative Learning Control design without a priori knowledge of the control direction. *Automatica*, 40:1803–1809, 2004.

H. Yue and H. Wang. Minimum entropy control of closed-loop tracking errors for dynamic stochastic systems. *Automatic Control, IEEE Transactions on*, 48(1):118–122, 2003. ISSN 0018-9286.

H. Yue, J. Zhou, and H. Wang. Minimum entropy of b-spline pdf systems with mean constraint. *Automatica*, 42(6):989–994, June 2006.