ISSN : 0974 - 7435 Volume 10 Issue 20

2014

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(20), 2014 [11883-11891]

Data driven PID-type neural network controller design using lazy learning for CSTR

Hongcheng Zhou [1, 2 *], Daobo Wang [1], Dezhi Xu [3], Qiang Zhang [4]

[1]College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing Jiangsu, 210016, (CHINA)

[2]Institute of Information, JinLing Institute of Technology, Nanjing Jiangsu, 211169, (CHINA)

[3]Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Institute of Automation, Jiangnan University, Wuxi Jiangsu, 214122, (CHINA)

[4]School of Electrical Engineering,University of Jinan, Jinan Shandong, 250022, (CHINA)

E-mail : Hongcheng Zhou, Email:zhouhc8@163.com

ABSTRACT

Since most chemical processes exhibit severe nonlinear and time-varying behavior, the control of such processes is challenging. In this paper, a novel two-layer online adjust algorithm is presented for chemical processes. The lower layer consists of a conventional PID-type neural network (PIDNN) controller and a plant process, while the upper layer is composed of identification and tuning modules. Using a lazy learning algorithm, a local valid linear model denoting the current state of system is automatically exacted for adjusting the PID controller parameters based on input/output data. This scheme can adjust the PIDNN parameters in an online manner even if the system has nonlinear properties. In this online tuning strategy, the BP training algorithm is considered. The simulation results on the dynamic model of Continuous Stirred Tank Reactor (CSTR) are provided to demonstrate the effectiveness of the proposed new control techniques.

KEYWORDS

CSTR, PID, Neural network, Data driven, Lazy learning.

INTRODUCTION

With the development of control theory and technology, many scholars put forward the adaptive online adjustment strategy, such as: based on the optimization strategy parameters self-tuning, based on the generalized minimum variance parameter adjustment, etc. [1-2]. When using these advanced strategy, it requires to identify the current system conditions of the model, which has two kinds of online and offline. In actual application of these strategies achieve good results. And in the face of a wide range, strong nonlinear, or jumping change system, if we adopt the rolling strategy, it tend to throw away the past data, relying only on the current some of the data obtained, can't let the system to establish a more accurate model, it is difficult to achieve good control effect [3-5]. If the offline global model is set up, because the new point, often need to be trained, the offline model is set up again, big workload.

Continuous stirred tank reactor (CSTR) is a kind of chemical reactor. Due to its low cost, strong heat transfer capability, stable product quality characteristics, it becomes the core equipment of producing polymer. It has been widely used in industrial production process [6-7]. Continuous stirred reactor control variables including pressure, temperature, and concentration and so on, the control of these variables will directly affect the quality of the product. In actual production process, continuous stirred reactor would be affected by many unfavourable factors. Therefore, it is difficult to establish precise mathematical model. It is not easy to achieve performance control. So, the PID controller plays an important role. However, nonlinear systems tend to have stronger uncertainty, time-varying.

Using conventional PID controller is difficult to achieve ideal control effect. The PID controller parameter setting often need the rich experience and repeated attempts. It is not necessarily that can achieve goal, this limits the PID controller used widely. And the actual system is often accompanied by a lot of online and history data. How to adjust the parameters of the PID controller has attracted extensive attention.

In order to overcome the above problem, we use the real-time algorithm adaptive, the nature of the K-VNN search strategy, accumulated data to find out from the system and the current point matching the data set, a local polynomial fitting method is adopted to establish the local model of the system. The K-VNN method has be introduced to lazy learning method [8-9]. And based on tracking error of the least performance index, is deduced with the PID control law in the form of structure, so as to realize the adaptive PID control for nonlinear systems.

The rest of this paper is organized as follows. Section 2, the lazy learning modelling method is given. In Section 3, main results of adaptive PID-type neural network (PIDNN) control are proposed. Simulation results are presented to show the effectiveness of the proposed technique in Section 4. Finally, some conclusions are made at end of this paper.

MODELLING METHOD (LAZY LEARNING)

Lazy learning algorithm based on "input to produce output of similitude" principle. It is generally the sample data memory in memory, and then according to the input point, find similar data in the sample data, according to the sample data to get the input to the corresponding output. Therefore, it is also called "Based on the study of Memory" (the MBL: Memory -based Learning). Describe the input data and sample data correlation criterion generally USES the distance function, which recently with the input point data has the higher correlation.

Considering an unknown nonlinear mapping $f : R^n \to R$, we assumed that can get system observable input and output data $\left\{\left(X_i, y_i\right)\right\}_{i=1}^{N}$. And this set data has function relation:

$$y_i = f\left(x_i\right) + \varepsilon \tag{1}$$

where the independent variable $X \in R^n$, the dependent variable $y_i \in R$, zero mean $\varepsilon_i \in R$ and variance σ^2 for the independent random distribution variables. The question is whether any vector X_q of the input space can according to the existing data sets create a mapping. It can pass the mapping and get the system corresponding estimate output $\hat{y}_q$. This problem can be summed up in solving the following optimization problem.

$$\min \sum_{(X_i, Y_i) \in \Omega_k} \left(y_i - f\left(X_i, \theta\right)\right)^2 w_i \tag{2}$$

where Ω_k is the local space X_k of the nearest k samples. $f\left(\cdot\right)$ describes the nonlinear mapping function of the input and output vector. w_i is weight value, it is local space within the sample data to the influence degree of the output vector. Different sample data in local space effect on the system output is different. Intuitively, the closest to the input vector corresponding to the sample vector output can most reflect the current output. This is also the basic principle of lazy learning algorithm: analogy input produces analogy output.

By using the principle of the above algorithm, we assume that the controlled system available can use NARX model to

represent as follows:

$$y(t) = f(\phi(t-1)) + \varepsilon(t) \quad \textbf{(3)}$$

where

$$\phi(t-1) = \left[y(t-1), \cdots, y(t-n_y), u(t-p), \cdots, u(t-p-n_u) \right]^T$$

It is system state vector on t moment. $m(m = n_y + n_u); n_y, n_u, p > 0$ is the system output, the system input order and the system delay. $y(t)$, $u(t)$, $\varepsilon(t)$ are the system output, the system input, the zero mean white noise. $f(\cdot)$ is unknown nonlinear function.

About the system (3) the physical description of the unknown, assuming that there are N sets of input and output data $\{y(i), \phi(i)\}_{i=1}^{N}$ at the current time t, system input information $\phi(t)$, the K-VNN search strategy, in the system of existing N sets of data to find the most similar data (K<< N), specific as follows:

When $\cos\beta(\phi(i), \phi(t)) < 0$, we think $\phi(i)$ diverge from the current input $\phi(t)$, which is unfavourable for system modelling, so we should discard the data;

Otherwise, with the index nuclear of $\phi(i)$, $\phi(t)$ and the angle cosine weighted sum, namely

$$D(\phi(i), \phi(t)) = \alpha \cdot e^{-d(\phi(i), \phi(t))} + (1-\alpha) \cdot \cos\beta(\phi(i), \phi(t)) \quad \textbf{(4)}$$

Where:

$$\begin{cases} \cos\beta(\phi(i), \phi(t)) = \dfrac{\phi^T(i)\phi(t)}{\|\phi(i)\|_2 \cdot \|\phi(t)\|_2} \\ d(\phi(i), \phi(t)) = \|\phi(i) - \phi(t)\|_2, \alpha \in [0,1] \end{cases}$$

By (2), the weighted selection criteria $D(\phi(i), \phi(t))$ directly reflects $\phi(i)$ and $\phi(t)$ similarity. If the two vectors are the closer, d is the smaller, and $\cos\beta$ is the greater, thereby $D(\phi(i), \phi(t))$ is bigger. So in the existing data information, (2) selects the largest k set data values $D(\cdot)$, in descending order, constructed learning sets:

$$\begin{aligned} &\{(\phi(1), y(1)), \cdots, (\phi(k), y(k))\} \\ &D(\phi(1), \phi(t)) > \cdots > D(\phi(k), \phi(t)). \end{aligned} \quad \textbf{(5)}$$

The information vector is linear regression; the system can get the local linear model of the current conditions. But at different working conditions, in accordance with the current point data $\phi(t)$ may not be the same density. The data number used in modelling is also uncertain, namely: modelling neighbourhood values are variable, in order to get the best linear model $\hat{\theta} = \left[d, a_1, \cdots, a_{n_y}, b_0, b_1, \cdots, b_{n_u} \right]^T$, reduce the amount of calculation, set up neighbourhood range $k \in [k_m, k_M] (k_m < k_M)$ in advance. When calculating the local model $\hat{\theta}_{k+1}$ of the neighbouring $k+1$, we directly use model values θ_k of the neighbour k, and use the recursive least squares method, so the system model can be get as

$$\begin{cases} \psi_{k+1} = \Phi_{k+1}^T P_k \Phi_{k+1} + 1 / D_{k+1}, \\ \gamma_{k+1} = P_k \Phi_{k+1} \psi_{k+1}^{-1}, \\ e_{k+1} = y(k+1) - \Phi_{k+1}^T \hat{\theta}_k, \\ \hat{\theta}_{k+1} = \hat{\theta}_k + \gamma_{k+1} e_{k+1}, \\ P_{k+1} = P_k - \gamma_{k+1} \gamma_{k+1}^T \psi_{k+1}. \end{cases} \quad \textbf{(6)}$$

We get the model θ_{k+1} of the near $k+1$. At the same time, we can get a cross error value of the near $k+1$,

$$e_{k+1,j}^{loo} = y(j) - \Phi_j \hat{\theta}_{k+1}^{-j} = \frac{y(j) - \Phi_j^T \hat{\theta}_{k+1}}{1 - \Phi_j^T P_{k+1} \Phi_j}, \; j = 1,2 \cdots; k+1. \quad \textbf{(7)}$$

In $k+1$ sets data, it represented local model $\hat{\theta}_{k+1}^{-j}$ by removing the jth data. $e_{k+1,j}^{loo}$ is the error value between the actual value $y(j)$ and the model $\hat{\theta}_{k+1}^{-j}$. In this way, we can get cross error $\left\{ e_{k+1,j}^{loo} \right\}_{j=1}^{k+1}, k+1 \le k_M$, mean square, and $e_{k+1,j}^{loo}$ of the neighbour $k+1$

$$E^{loo}(k+1) = \frac{\sum_{j=1}^{k+1} w_j \left(e_{k+1,j}^{loo} \right)^2}{\sum_{j=1}^{k+1} w_j}. \quad \textbf{(8)}$$

where the weighted factor $w = \sqrt{D\left(\phi(j), \phi(t)\right)}$ directly reflect each $\phi(j)$ the size of the cross error $E^{loo}(k+1)$. The $\phi(j)$ is closer $\phi(t)$, its contribution is the greater, whereas the smaller.

$$E^{loo}(k+1) > E^{loo}(k), k+1 \in \left[k_m, k_M \right]. \quad \textbf{(9)}$$

From (9), we think model variation, then stop the regression calculation, and use the model θ_k as the system optimal model of the current time. Otherwise, by using the recursive least squares algorithm model, we select new information vector from learning sets, and continue to iteration, until $k = k_M$. So, we can judge the merits of the local model in time, get in line with the current moment system input and output relationship of the best local linear model.
At each sample point, the nonlinear system can be represented by a locally valid linear model in a certain region around this point

$$\hat{y}(t) = \phi(t-1)^T \hat{\theta} \quad \textbf{(10)}$$

From above (10), we can obtain the Jacobian of plant (3), that is $\frac{\partial y(t)}{\partial u(t)} \approx \frac{\partial \hat{y}(t)}{\partial u(t)} = b_0$.

ADAPTIVE PID-TYPE NEURAL NETWORK CONTROL VIA LAZY LEARNING

In this section, a novel two-layer online tuning strategy has been developed in this paper and the control structure is shown in Figure 1. The lazy learning is derived to model the nonlinear plant, and then a PIDNN is designed to control it. The control objective is to tune the parameters in the PIDNN controller so that the output $y(t)$ of the nonlinear system tracks the desired trajectories $y_d(t)$, at the same time, to tune the parameters in the lazy learning identifier so that its output $\hat{y}(t)$ can model the system nonlinear behaviour $y(t)$ in real time. Therefore, the lazy learning identifier can provide updated model information to the PIDNN controller every time step.

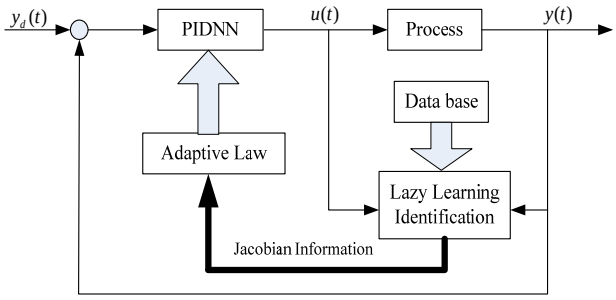

Figure 1 : Control system framework

PID controller

A typical discrete-time PID controller can be expressed as,

$$u(t) = K_P(t) \cdot e(t) + K_I(t) \cdot \sum e(t) + K_D(t) \cdot \Delta e(t) \quad \textbf{(11)}$$

where $u(t)$ is the control effort at time t; $K_P(t)$, $K_I(t) = \frac{K_P(t)T_s}{T_i}$ and $K_D(t) = \frac{K_P(t)T_d}{T_s}$ are proportional, integral and derivative gains, respectively; T_s represents the sampling time; T_i and T_d represent the integral and derivative constants, respectively, $e(t)$ is the tracking error defined as $e(t) = y_d(t) - y(t)$, and $\Delta e(t) = e(t) - e(t-1)$; $y_d(t)$ is the desired plant output, $y(t)$ is the actual plant output.

The PID controller Eq. (27) can also be expressed in the following incremental form,

$$\Delta u(t) = u(t) - u(t-1) = \sum_{i=1}^{3} K_i(t) \cdot e_i(t) \quad \textbf{(12)}$$

where $K_i(t)$ for $i = 1,2,3$ are the parameters of PID controller, $e_1(t) = e(t) - e(t-1)$, $e_2(t) = e(t)$ and $e_3(t) = e(t) - 2e(t-1) + e(t-2)$. In general, if the parameters $K_i(t)$ are chosen to be optimal, a satisfied tracking performance could be obtained. However, it is difficult to select these parameters due to the plant nonlinearity. In this paper, a neural network is used to formulate the PID controller, the weights in which are corresponding with the parameters of PID controller, so that the parameters $K_i(t)$ can be updated through online training of the neural network.

PID-type neural network (PIDNN)

As shown in Figure 1, the PIDNN acts as a controller, the output of the first layer net_j and the output of the network $\Delta u(t)$ are given as

$$net_j = \sum_{i=1}^{3} v_{ij} \cdot e_i(t) \quad \textbf{(13)}$$

$$u(t) = u(t-1) + \Delta u(t) = u(t-1) + \sum_{j=1}^{3} K_j(t) \cdot h(net_j) \quad \textbf{(14)}$$

where v_{ij} for $i = 1,2,3$ and $j = 1,2,3$ are the weights between the input layer and the hidden layer, and $h(\cdot)$ is a nonlinear activation function of hidden layer, which can be selected as

$$h(net_j) = \frac{1 - e^{-net_j}}{1 + e^{-net_j}} \quad \textbf{(15)}$$

Thus, the derivative of Eq. (15) is,

$$h'(net_j) = \frac{1}{2}\left(1 - h(net_j)\right)^2 \quad \textbf{(16)}$$

The training algorithm can now be formulated in order to make the PIDNN adaptive.

Training algorithm

For the adaptive PIDNN controller, its weights are updated along the negative gradient of a given error function as follows,

$$J_C(t) = \frac{1}{2}e^2(t) = \frac{1}{2}\left(y_d(t) - y(t)\right)^2 \quad \textbf{(17)}$$

The gradients of $J_C(k)$ with respect to weights $K_j(k)$ and v_{ij} can be evaluated as

$$\frac{\partial J_C(t)}{\partial K_j(t)} = -e(k)\frac{\partial y(t)}{\partial K_j(t)} = -e(t)\frac{\partial y(t)}{\partial u(t)}h(net_j) \quad \textbf{(18)}$$

$$\frac{\partial J_C(t)}{\partial v_{ij}(t)} = -e(k)\frac{\partial y(t)}{\partial v_{ij}(t)} = -e(t)\frac{\partial y(t)}{\partial u(t)}K_j(t)h'(net_j)e_i(t) \quad \textbf{(19)}$$

In Eqs. (18) and (19), $\frac{\partial y(t)}{\partial u(t)}$ denotes the sensitivity of the plant output with respect to its input, namely Jacobian matrix. The sensitivity cannot be calculated directly from the output of the nonlinear system since the precise mathematical model is usually unknown in many practical systems. In general, since the lazy learning was trained (off-line or online), we can obtain the following by (10). Therefore, the system sensitivity can be calculated as

$$y_u(k) = \frac{\partial y(t)}{\partial u(t)} \cong \frac{\partial \hat{y}(t)}{\partial u(t)} = b_0 \quad \textbf{(20)}$$

Using a BP training algorithm, the weights adaptation laws for the output and hidden layers are derived as

$$K_j(t+1) = K_j(t) - \eta_c \cdot \frac{\partial J_C(t)}{\partial K_j(t)},\ j = 1,2,3 \quad \textbf{(21)}$$

$$v_{ij}(t+1) = v_{ij}(t) - \eta_c \cdot \frac{\partial J_C(t)}{\partial v_{ij}(t)},\ i,\ j = 1,2,3 \quad \textbf{(22)}$$

where the respective partial derivatives in Eqs. (21) and (22) are shown in Eqs. (18) and (19).

THE SIMULATION RESULTS

Continuous stirred tank reactor (CSTR) is an important unit of chemical process. It has the strong nonlinear characteristics, and it is a typical nonlinear object in chemical system. Dynamic equations for the input and output are

$$\begin{cases} \dfrac{\mathrm{d}C_a}{\mathrm{d}t} = \dfrac{q}{V}\left(C_{af} - C_a\right) - k_0 C_a \exp\left(-E/RT\right), \\ \dfrac{\mathrm{d}T}{\mathrm{d}t} = \dfrac{q}{V}\left(T_f - T\right) + \dfrac{\left(-\Delta H\right)k_0 C_a}{\rho C_p}\exp\left(-E/RT\right) \\ \qquad\qquad + \dfrac{\rho_c C_{pc}}{Q_c C_p}Q_c\left[1 - \exp\left(\dfrac{-hA}{Q_c \rho_c C_{pc}}\right)\right]\left(T_{cf} - T\right). \end{cases}$$

Here the density C_a of the product A in the reactor, the reactants temperature T, the in and out of the material flow q, the coolant flow Q_c, the coolant inlet temperature T_{cf}, the coolant outlet temperature T_c, the feed concentration C_{af}, the feed temperature T_f. By adjusting the size of the coolant flow, we can control the temperature T in the reactor, thus controlling the reactants concentration C_a. According to process requirements, we determine the process output C_a and T. Where: the control variables Q_c, the external disturbance variables T_c, q, C_{af} The physical parameters are shown in TABLE 1.

TABLE 1 : Model parameters

Parameter	Value
q	100 L/min
T_f	350 K
V	100 L
k_0	7.2×10^{10} L/min
$-\Delta H$	2×10^{5} cal/mol
C_p, C_{pc}	$1\ \text{cal}/(\text{g}\cdot\text{k})$
T_0	440.2 K
T_s	0.1 min
C_{af}	1.0 mol/L
T_{cf}	350 K
hA	7×10^{5} cal/$(\text{min}\cdot k^{-1})$
E/R	9.95×10^{3} K
ρ,ρ_c	1000 g/L
Q_{c0}	103.41 L/min
C_{a0}	0.0836 mol/L

In the table, a static working point of CSTR corresponding steady-state value is C_{c0}, T_0, C_{a0}. In the static working point, when it changes $\pm20\%$ of the coolant flow $Q_{c0}=103.41\ L/\text{min}$, it will produce 2000 set samples.

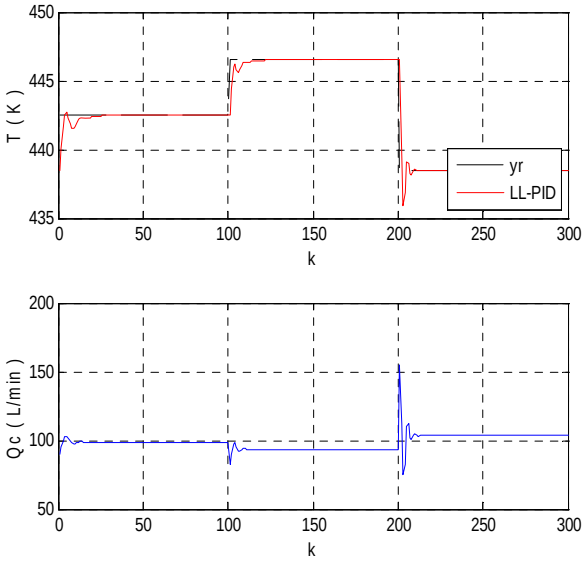

Figure 2 : The CSTR response curve

Input vector of the model is:

$$\phi(t-1)=\left[T(t-1),T(t-2),Q_c(t-1),Q_c(t-2)\right]$$

The local model order is $n_y=2$, $n_u=1$. Lazy learning online modelling parameters are $\alpha=0.85$, $k\in[12,100]$. Punishment of PID controller is $Q=9.3$. Output variable change value $T(446.5\rightarrow444.5\rightarrow438.5)$ is shown in Figure 2.

Figure 2 shows that when the system working point move, lazy learning method choice of modelling data due to the nature of time and space adjacent to the change of the fast tracking system, in order to gain parameters of PID controller has better adaptive. In the whole tracking trajectory, PID control parameters are shown in Figure 3.

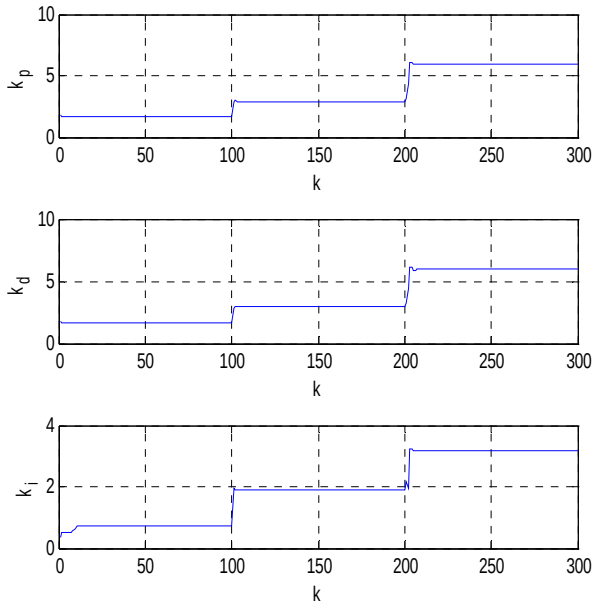

Figure 3 : The change curve of PIDNN controller parameters

CONCLUSIONS

We have carried out a systematic study on the data driven PIDNN control of a CSTR in this paper. The lazy learning algorithm, a local valid linear model denoting the current state of system is automatically exacted for adjusting the PIDNN controller parameters based on input/output data. This scheme can adjust the PIDNN parameters in an online manner even if the system has nonlinear properties. Finally, simulation results are provided on CSTR to show the effective and advantages of the control strategy.

ACKNOWLEDGMENTS

This work was supported by Qing Lan Project of Jiangsu province, 333 Project of Jiangsu province, Aeronautical Science Foundation of China (No. 20111052010), the National 863 Program (No. 2007AA01Z404), Universities Natural Science Research Project of Jiangsu Province (No. 13KJD520005), Modern Educational Technology in Jiangsu province (2013-R-26144). This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61403161.

REFERENCES

[1] M.Xu, S.Y.Li, W.J.Cai; Receding horizon optimization approach to PID controller parameters auto tuning, Acta Automatica Sinica, **31(3)**, 459 -463 **(2005)**.

[2] J.H.Chen, T.C.Huang; Applying neural networks to online updated PID controller for nonlinear process control, Journal of Process Control, **14(2),** 211-230 **(2004)**.

[3] Y.Zhang, Z.Q.Chen, Z.Z.Yuan; Nonlinear system PID-type multi-step predictive control, Journal of Control Theory & Applications, **21(2)**, 201-204 **(2004)**.

[4] D.Xu, B.Jiang, P.Shi; A novel model free adaptive control design for multivariable industrial processes, IEEE Trans. on Industrial Electronics, **61(11)**, 6391-6398 **(2014)**.

[5] D.Xu, B.Jiang, P.Shi; Adaptive observer based data-driven control for nonlinear discrete-time processes, IEEE Trans. on Automation Science and Engineering, **11(4)**, 1037-1045 **(2014).**

[6] W.Wang; Research on Control Methods of CSTR, Beijing Jiaotong university, **(2013)**.

[7] T.Wang; Research on soft sensing methods for estimating parameters of the concentration in continuous stirred tank reactor, Northeastern University, **(2008)**.

[8] G.Bontempi, M.Birattari; From linearization to lazy learning: a survey of divide-and-conquer techniques for nonlinear control, International Journal of Computational Cognition, **3(1),** 56 -73 **(2005)**.

[9] W.Sun, W.Wang, R.Zhu; Iterative learning control for nonlinear system using lazy learning method, Control and Decision, **3(18)**, 263-266 **(2003)**.