Journal of Marine Science and Technology, Vol. 28, No. 5, pp. 357-366 (2020)
DOI: 10.6119/JMST.202010_28(5).0005

ADAPTIVE PID-LIKE CONTROL USING BROAD LEARNING SYSTEM FOR NONLINEAR DYNAMIC SYSTEMS

Ching-Chih Tsai[1], Chun-Chieh Chan[1], Chien-Cheng Yu[2], Hung-Sheng Chen[1], Guo-Shun Hung[1]

Key words: broad learning system, PID-Like control, nonlinear dynamic systems, Lyapunov stability theory.

ABSTRACT

This paper presents a new learning control structure using broad learning system (BLS) for adaptive PID-like control of unknown digital nonlinear dynamic systems with time delays. The proposed control method, abbreviated as BLS-APIDLC, is novel in combining BLS and model predictive control to develop a new PID-like control law for high-performance setpoint tracking control and disturbance rejection. Comparative simulations on two renowned nonlinear digital time-delay systems are well used to show the effectiveness and superiority of the proposed method by comparing to four existing methods.

INTRODUCTION

In the past decades, conventional proportional-integral-derivative (PID) controllers have gained widespread use in numerous control applications due to its simplicity of design and efficiency in the industrial applications (Astrom and Hagglund, 1995; Silva et al., 2005; O'Dwyer, 2009; Vilanova and Visioli, 2012). Although the PID controller has only three parameters to be tuned, it is surprisingly difficult to find the right tuning for them without systematic procedures. As such, the tuning of the PID gains is always a challenge in the state of the art of PID controller design. In other words, the main problem with a PID controller is the fact that the parameters of the PID controller must be adjusted properly to meet desired performance. This problem becomes more important when considering issues that include stability and control performance. Recently, the area of adaptive PID control and its related control approaches have still been developed by researchers (Oliveira and Lemos, 2000; Pan et al., 2007; Fahmy et al., 2014; Yang et al., 2015). Adaptive PID controller designs with controller parameters updated online by the neural network models were also presented. Pan et al. (2007) developed a two-layer supervised control method for tuning PID controller parameters based on model parameters estimated by the lazy learning technique. Fahmy et al. (2014) proposed an adaptive PID controller using the recursive least square algorithm which updates the PID gains automatically online to force the actual system to behave like a desired reference model. Oliveira and Lemos (2000) proposed a comparison of some fuzzy-model-based adaptive-predictive control strategies. Yang et al. (2015) presented an adaptive predictive control strategy based on Laguerre functions in the chopper cascade control system, and examined by experiments.

More recently, machine learning algorithms have made significant progress, especially deep learning technologies that have been made in wide areas (Tsai et al., 2014; Rosa and Yu, 2016; Ghazia et al., 2017; Andò et al., 2018; Y. Kang et al., 2019). By successively adjusting the weights between neurons over many input-output pairs, the function computed by the network is refined over time so that it provides more accurate predictions. The lately presented broad learning system (BLS) is an emerging way for efficient and effective modeling of complex systems (Chen and Liu, 2018; Jin and Chen, 2018; Chen et al., 2019). Chen and Liu (2018) developed a very fast and efficient BLS based on the random vector functional-link neural networks (RVFLNN) (Pao et al., 1994) to offer an alternative way for deep learning and structure. The designed model can be expanded in wide fashion when new feature nodes and enhancement nodes are needed. Moreover, the corresponding incremental learning algorithm is also designed. The BLS offers an alternative to deep learning because it has a fast and broad expansion without the need for retraining through incremental learning. The input signals are passed to

Paper submitted 01/12/20; revised 03/25/20; accepted 05/03/20. Author for correspondence: Ching-Chih Tsai (e-mail: cctsai@nchu.edu.tw).
[1] Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan.
[2] Department of Electronic Engineering, Hsiuping University of Science and Technology, Taichung, Taiwan.

the mapped feature layer and then passed to the enhancement layer via a nonlinear transformation. Although NNs possess good function approximation capabilities for nonlinear dynamic systems, the training process is time-consuming. On the other hand, the BLS system has been shown to preserve good function approximation capabilities and has been illustrated the feasibility and benefits of BLS-based control techniques for identification and control of nonlinear dynamic systems (Chen and Liu, 2018; Jin and Chen, 2018; Vong et al., 2020; Xu et al., 2018; and Feng and Chen, 2018a; Chen et al., 2019).

Conventional PID controllers have been regarded as the simplest and the most deployed controller in industry. To extend the robustness and adaptability of the conventional PID controller, by integrating the simplicity and effectiveness of the conventional PID controller and the learning and automatic adjustment capabilities of the intelligent control strategy based on the PID-like controller for the nonlinear dynamic system have been proposed (Wang et al., 1997; Tsai et al., 2005; Cong and Liang, 2009; Fu and Chai, 2011). For example, Wang et al. (1997) presented an adaptive PID-like controller using a Modified Neural Network (MNN) for learning the characteristics of a dynamic system. Tsai et al. (2005) proposed an adaptive PID-Like fuzzy-neural controller and applied to the nonlinear model reference control system. Fu and Chai (2011) presented a robust self-tuning PID-like controller by combining a pole assignment self-tuning PID controller with a filter. Cong and Liang (2009) proposed a PID-like neural network nonlinear adaptive controller for motion control systems by using a mix locally recurrent neural network. The gradient descent method is used for online adjustment and the initial PID parameters are needed which can operate the closed-loop stably. Kumar et al. (2014) proposed a hybrid neural network-based PID like adaptive controller for precise position control of a permanent-magnet synchronous motor (PMSM) servo drive. So far, many adaptive PID control for industrial applications have been proposed (Tung, 2012; Tsai et al., 2017; Tsai et al., 2019). Feng and Chen (2018b) presented a PID-like control method using BLS; however, this kind of PID-like control method was limited to a class of nonlinear dynamic systems without time-delays.

Inspired by the above-mentioned surveys, the objective of this paper is to propose a BLS-based adaptive predictive PID-like control, called BLS-APIDLC, of a class of unknown nonlinear discrete-time time-delay systems not only for disturbance rejection but also for precise tracking and guaranteed stability. The presented contents of the paper are written in two principal contributions. One is the theoretical derivation and proof of a more general adaptive PID-like control approach using BLS for unknown nonlinear time-delay dynamic systems by comparing to the result (Feng and Chen, 2018b), and the other is comparative investigation of the proposed BLS-APIDLC in comparison with four existing adaptive PID control methods.

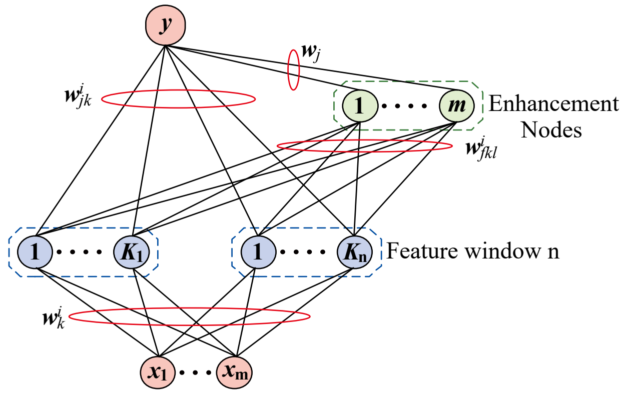

Fig. 1. Topological structure of the used BLS model.

The rest of this paper is organized as follows. The basic ideas of the BLS and BLS identifier design are described in Sections II and III, respectively. Section IV is devoted to proposing a BLS-APIDLC control law, investigating its closed-loop stability and iterative control algorithm. Section V uses computer simulations to explore the effectiveness and superiority of the proposed BLS-APIDLC method for two illustrious nonlinear time-delay systems. Section VI is finished with the conclusions and future work of the paper.

II. BROAD LEARNING SYSTEM (BLS)

1. Introduction to BLS

Fig. 1 illustrates the network structure of BLS, which is delineated as follows. First, the raw data is mapped into mapped features. Then, the mapped features are promoted as enhancement nodes with randomly generated weights. Finally, all mapped features and enhancement nodes are connected to the system output nodes. One defines the weight matrices connecting the outputs of mapped feature nodes and enhancement nodes to the system output nodes as

$$\boldsymbol{W} = [W_{fi} \;\; \beta_{fi} \;\; W_{ef} \;\; \beta_{ef} \;\; W_{ye} \;\; \beta_{ye}]^T \qquad (1)$$

where $W_{fi} = [w_{fkl}^i]_{M \times K_i}$, $w_{f_{kl}}^i$ is the weight connecting the l-th input x_l to the k-th feature node in the i-th mapping group; $\beta_{fi} = [b_{fk}^i]_{1 \times K_i}$, $b_{f_k}^i$ is the bias term associated to the k-th feature node in the i-th mapping group; $W_{ef} = [w_{jk}^i]_{(K_1+K_2+\cdots+K_n)\times m}$, w_{jk}^i is the weight connecting the k-th feature node of the i-th mapping group to the j-th enhancement node; $\beta_{ef} = [b_j]_{1\times m}$, b_j is the bias term associated to the j-th enhancement node; $\boldsymbol{W}_{ye} = [w_1^1, \cdots w_{K_1}^1, \cdots, w_n^n, \cdots w_{K_n}^n, w_1, \ldots, w_m]^T$ is the weight matrix connecting the outputs of feature nodes and enhancement nodes to the system output nodes, w_k^i is the weight connecting the k-th feature node in the i-th mapping group to the output neuron, and w_j is the weight connecting the j-th enhancement node to the output node. It is worth noting that all the weights and biases, w_{fkl}^i and b_{fk}^i, in the feature mapping layer can be specified either by a sparse auto-encoder or by an iterative learning algorithm.

2. System Identification and Control Using BLS

This subsection is devoted to briefly describing how the BLSs can be used to achieve system identification and control. The basic idea in doing so is to find on-line learning algorithms to update the key parameters of used BLSs so as to minimize a kind of quadratic cost function, $E(k)$. A basic gradient-based technique can be easily used to minimize the cost function. This results in the following rule for the adaptation of model parameters:

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - \eta\, \partial E(k) / \partial \boldsymbol{W}(k) \tag{2}$$

where $\eta > 0$ is the learning rate, and $\boldsymbol{W}(k)$ is the weight vector containing all model parameters at the time instant k, $\boldsymbol{W}(k)$ is the updating parameter vector, and $\partial E(k)/\partial \boldsymbol{W}(k)$ is the gradient of $E(k)$ with respect to $\boldsymbol{W}(k)$. It has been shown that such a learning procedure minimizes the global cost function $E(k)$ provided that the learning rate η is sufficiently small [5].

III. BLS INDENTIFIER

This section is dedicated to delineate a BLS identifier which is used to learn the input-output relationship of the nonlinear discrete-time time-delay system. Fig. 2 illustrates the conceptual block diagram of the BLS identifier. To explain how the BLS identifier works, let us describe the digital plant model with time-delay d by using the subsequent delayed form.

$$y(k) = f\begin{pmatrix} y(k-1), y(k-2), \cdots, y(k-n_y), \\ u(k-d), u(k-d-1), \cdots, u(k-d-n_u) \end{pmatrix} \tag{3}$$

where the plant input as $u(\cdot): Z^m \to \Re^1$, the plant output as $y(\cdot): Z^m \to \Re^1$, $f(\cdot): \Re^{n_y+n_u-d+1} \to \Re^1$ be a nonlinear function, $n_y \in Z^+$, $n_u \in Z^+$, $d \in Z^+$ represents the known time-delay of the system. Moreover, the incremental plant model with the time-delay d can be rewritten by the following advanced form.

$$y(k+d) = g\begin{pmatrix} y(k+d-1), y(k+d-2), \cdots, y(k+d-n_y), \\ \Delta u(k), \Delta u(k-1), \cdots, \Delta u(k+d-n_u) \end{pmatrix} \tag{4}$$

1. Updating Laws for the BLS Identifier

In the subsection, we will propose an iterative learning algorithm for BLS to approximate the nonlinear system in (4).

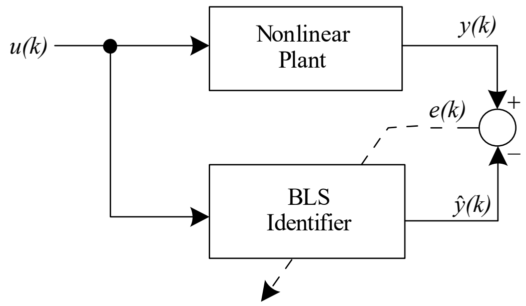

Fig. 2. Conceptual diagram of the BLS identifier.

Suppose there are n mapping groups with K_i feature nodes in the i-th group, and there are one group of m enhancement nodes. One denotes the weight matrix connecting the outputs of feature nodes and enhancement nodes to the output node as in (4). The input vector is denoted by, $\boldsymbol{X}_l$, $l = 1, \cdots, m$, and the BLS output vector $\hat{\boldsymbol{y}}$ is given as

$$\hat{\boldsymbol{y}}(\boldsymbol{k}) = \sum_{i=1}^{n}\sum_{k=1}^{K_i} \bar{w}_k^i F_k^i + \sum_{j=1}^{m} \bar{w}_j \xi_j \left(\sum_{i=1}^{n}\sum_{k=1}^{K_i} \bar{w}_{jk}^i F_k^i + \bar{b}_j \right) \tag{5}$$

where $F_k^i = \phi_k^i \left(\sum_{l=1}^{M} \left(\bar{w}_{f_{kl}}^i x_l + \bar{b}_{f_k}^i \right) \right)$ denotes the output of the k-th feature node in the i-th mapping group, and $\phi_k^i(x) = x$ is a linear activation function, and $\xi_j(x) = \tanh(x)$ is a nonlinear activation function.

The proposed iterative learning algorithms are based on the gradient descent method; the algorithms first find the gradients of all the updating parameters in the used BLS identifier as shown in Fig. 2, and then use the deepest descent approach to obtain the iterative parameter learning rules or algorithms. In doing so, to obtain the learning algorithms for the six types of parameters, $\bar{w}_{f_{kl}}^i$, $\bar{b}_{f_k}^i$, $\bar{w}_{jk}^i$, $\bar{b}_j$, $\bar{w}_k^i$ and $\bar{w}_j$, of the BLS identifier in (5), one defines the error function $E(k)$ for the BLS identifier by

$$E_I(k) = \frac{1}{2}(y(k) - \hat{y}(k))^2 = \frac{1}{2} e_I^{\,2}(k) \tag{6}$$

where $\hat{y}(k)$ denotes the output of the BLS identifier. By utilizing the deepest gradient descent algorithm, one obtains the learning law for the parameter vector $\boldsymbol{W}_I$ via (2).

$$\boldsymbol{W}_I(k+1) = \boldsymbol{W}_I(k) + \eta_I e_I(k) \frac{\partial \hat{y}(k)}{\partial \boldsymbol{W}_I(k)} \tag{7}$$

where η_I is the real and positive learning rate for the identifier, $\boldsymbol{W}_I = [W_{fi}^I \;\; \beta_{fi}^I \;\; W_{ef}^I \;\; \beta_{ef}^I \;\; W_{ye}^I \;\; \beta_{ye}^I]^T$ and

$$\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}=\left[\frac{\partial\hat{y}(k)}{\partial W_{fi}^I(k)}\ \frac{\partial\hat{y}(k)}{\partial \beta_{fi}^I(k)}\ \frac{\partial\hat{y}(k)}{\partial W_{ef}^I(k)}\ \frac{\partial\hat{y}(k)}{\partial \beta_{ef}^I(k)}\ \frac{\partial\hat{y}(k)}{\partial W_{ye}^I(k)}\ \frac{\partial\hat{y}(k)}{\partial \beta_{ye}^I(k)}\right]^T$$

2. Updating Laws for the BLS Identifier

The following theorem will show that the proposed algorithm is asymptotically convergent via the discrete-time Lyapunov stability theory.

Theorem 1: The BLS identifier (5) with parameter learning rule (7) and persistently exciting inputs is asymptotically convergent provided that $\eta_I(k)$ satisfies the following condition

$$0<\eta_I<2/\max_k\left\|\left(\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right)\right\|_2^2 \tag{8}$$

Proof: To obtain the sufficient condition of the asymptotical convergence of the proposed BLS identifier with and persistently exciting inputs, let us define the Lyapunov function as

$$L_I(k)=(y(k)-\hat{y}(k))^2=e_I^2(k) \tag{9}$$

Then, the time difference of the Lyapunov function is given by

$$\Delta L_I(k)=L_I(k+1)-L_I(k)=\Delta e_I(k)\cdot\left[2e_I(k)+\Delta e_I(k)\right] \tag{10}$$

where

$$\begin{aligned}\Delta e_I(k)&=\left(\frac{\partial e_I(k)}{\partial \boldsymbol{W}_I(k)}\right)^T\Delta \boldsymbol{W}_I(k)=-\left(\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right)^T\Delta \boldsymbol{W}_I(k)\\&=-\eta e_I(k)\left(\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right)^T\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}=-\eta e_I(k)\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right\|_2^2\end{aligned} \tag{11}$$

and

$$\begin{aligned}\left\|\left(\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right)\right\|_2^2&=\sum_{i=1}^{n}\sum_{k=1}^{K_i}\left(\left(\frac{\partial\hat{y}(k)}{\partial \bar{w}_{fkl}^i}\right)^2+\left(\frac{\partial\hat{y}(k)}{\partial \bar{b}_{fk}^i}\right)^2\right)\\&+\sum_{j=1}^{m}\left(\left(\frac{\partial\hat{y}(k)}{\partial \bar{b}_j}\right)^2+\left(\frac{\partial\hat{y}(k)}{\partial \bar{w}_j}\right)^2\right)+\sum_{i=1}^{n}\sum_{k=1}^{K_i}\left(\left(\frac{\partial\hat{y}(k)}{\partial \bar{w}_k^i}\right)^2+\left(\frac{\partial\hat{y}(k)}{\partial \bar{w}_{jk}^i}\right)^2\right)\end{aligned} \tag{12}$$

Thus, from (10) and (11), it follows that

$$\begin{aligned}\Delta L_I(k)&=-\eta e_I(k)\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right\|_2^2\cdot\left[2e_I(k)-\eta e_I(k)\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right\|_2^2\right]\\&=-\eta e_I^2(k)\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right\|_2^2\cdot\left[2-\eta\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right\|_2^2\right]\end{aligned} \tag{13}$$

Obviously, $\Delta L_I(k)<0$ is negative definite if and only if the sufficient condition (8) holds. This completes the proof.

Theorem 2: The best asymptotical stability rate $\eta_I^*(k)$ for the BLS identifier is given by

$$\eta_I^*=1/\max_k\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right\|_2^2 \tag{14}$$

Proof: The best learning rate $\eta_I^*(k)$ can be found by considering at the time when $\max_k\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}(k)}\right\|_2^2$ occurs. Therefore, it follows that

$$\Delta L_I(k)=-e_I^2(k)\max_k\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right\|_2^2\cdot\left[2\eta_I-\eta_I^2\max_k\left\|\frac{\partial\hat{y}(k)}{\partial \boldsymbol{W}_I(k)}\right\|_2^2\right] \tag{15}$$

Hence, the best convergent rate in (14) can be easily obtained from (15) by solving $d\Delta L_I(k)/d\eta_I^*=0$.

Remark 1: In applying Theorem 1, it is required that the learning rate must be checked at every sampling time instant, in order to guarantee that the inequality (8) holds. Obviously, the least upper bound, $\max\left\|\partial\hat{y}(k)/\partial \boldsymbol{W}_I(k)\right\|$, is small when the learning phase just gets started, but the bound will be getting small and even becomes zero when the learning phase is achieved. Hence, it is practically safe to choose a small initial learning rate of $\eta_I(k)$, in order to ensure the uniform asymptotical convergence of the training BLS.

IV. BLS-BASED ADAPTIVE PID-LIKE CONTROLLER (BLS-APIDLC) DESIGN

This section presents a BLS-APIDLC method for a class of nonlinear discrete-time dynamic systems with time delays. The proposed controller is strictly derived and examined by integrating the previous BLS identifier with online learning and identification. A cost function of the BLS-PIDLC is also proposed, and the proposed controller can cope with parametric variation and uncertainty in the controlled plant through on-line learning. The controller's learning algorithm is considerably faster through the introduction of a BLS algorithm. Fig. 3 shows the detailed description of the BLS-APIDLC system using two BLS identifiers and one BLS-PIDLC for unknown nonlinear dynamic system models. In Fig. 3, both BLS identifiers have the same parameters, namely that all the parameters of the BLS identifier 2 directly copy from those of BLS identifier 1. Notice that the output of BLS identifier 2 is $\hat{y}(k+d)$ which is calculated by using the advanced form of the system model (5).

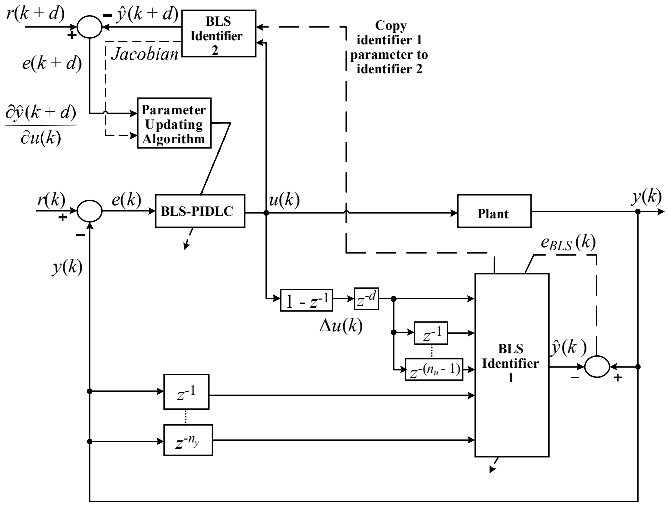

Fig. 3. Detailed block diagram of the BLS-APIDLC controller.

1. Design of the Proposed BLS-PIDLC

This subsection will propose iterative learning algorithms for the BLS-PIDLC in order to control the nonlinear dynamic system in (4). The proposed iterative learning algorithms for the BLS-PIDLC are almost similar to those for the BLS identifier, but the output of the BLS-PIDLC is given by

$$u(k)=\sum_{i=1}^{n}\sum_{k=1}^{K_i} w_k^i F_k^i+\sum_{j=1}^{m} w_j \xi_j\left(\sum_{i=1}^{n}\sum_{k=1}^{K_i} w_{jk}^i F_k^i+b_j\right) \quad (16)$$

The learning laws for the six kinds of parameters, $w_{f_{kl}}^i$, $b_{f_k}^i$, w_{jk}^i, b_j, w_k^i and w_j, of the BLS-PIDLC are derived to minimize the following predictive cost function

$$E_c(k+d)=\left(r(k+d)-\hat{y}(k+d)\right)^2/2=e_c^{\,2}(k+d)/2 \quad (17)$$

where $r(k+d)$ and $\hat{y}(k+d)$ respectively denote the d-step-ahead reference command and predictive output of the second BLS identifier. From the previous gradient descent algorithms, we have

$$\begin{aligned}\Delta \boldsymbol{W}_c(k)&=\boldsymbol{W}_c(k+1)-\boldsymbol{W}_c(k)\\&=\eta_c e_c(k+d)\frac{\partial \hat{y}(k+d)}{\partial u(k)}\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}\end{aligned} \quad (18)$$

where η_c is the real and positive learning rate for the BLS-PIDLC, $\boldsymbol{W}_C=[W_{fi}^C\ \ \beta_{fi}^C\ \ W_{ef}^C\ \ \beta_{ef}^C\ \ W_{ye}^C\ \ \beta_{ye}^C]^T$ and

$$\frac{\partial u(k)}{\partial \boldsymbol{W}_C(k)}=\left[\frac{\partial u(k)}{\partial W_{fi}^C(k)}\ \ \frac{\partial u(k)}{\partial \beta_{fi}^C(k)}\ \ \frac{\partial u(k)}{\partial W_{ef}^C(k)}\ \ \frac{\partial u(k)}{\partial \beta_{ef}^C(k)}\ \ \frac{\partial u(k)}{\partial W_{ye}^C(k)}\ \ \frac{\partial u(k)}{\partial \beta_{ye}^C(k)}\right]^T$$

Remark 2: In (18), the Jacobian operation, $\partial \hat{y}(k+d)/\partial u(k)$, can be online computed from the second BLS identifier whose weighting parameters are directly transferred from the first BLS identifier. Moreover, the computations regarding $\partial u(k)/\partial \boldsymbol{W}_c(k)$ are easily done via (16).

2. Asymptotical Stability of the Proposed BLS-PIDLC

The following theorem will prove that the proposed BLS-PIDLC control system is asymptotically stable.

Theorem 3: Assume that the parameters of both identifiers are convergent and the weighting parameter vector $\boldsymbol{W}_c$ in (16) is trained along with the BLS-PIDLC law via (18). Then the proposed BLS-PIDLC closed-loop system is shown uniformly asymptotically stable if the learning rate η_c satisfies the following condition

$$0<\eta_c<\frac{2}{\max_k\left(\frac{\partial \hat{y}(k+d)}{\partial u(k)}\right)^2\left\|\left(\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}\right)\right\|_2^2} \quad (19)$$

where

$$\begin{aligned}\left\|\left(\frac{\partial u(k)}{\partial \boldsymbol{W}(k)}\right)\right\|_2^2&=\sum_{i=1}^{n}\sum_{k=1}^{K_i}\left(\left(\frac{\partial u(k)}{\partial w_{fkl}^i}\right)^2+\left(\frac{\partial u(k)}{\partial b_{fk}^i}\right)^2\right)+\\&\sum_{j=1}^{m}\left(\left(\frac{\partial u(k)}{\partial b_j}\right)^2+\left(\frac{\partial u(k)}{\partial w_j}\right)^2\right)+\sum_{i=1}^{n}\sum_{k=1}^{K_i}\left(\left(\frac{\partial u(k)}{\partial w_k^i}\right)^2+\left(\frac{\partial u(k)}{\partial w_{jk}^i}\right)^2\right)\end{aligned} \quad (20)$$

Proof: To establish the sufficient condition of the uniformly asymptotical stability of the proposed BLS-PIDLC controller, we define the sequel Lyapunov function

$$L_C(k)=\left(r(k+d)-\hat{y}(k+d)\right)^2=e_c^2(k+d) \quad (21)$$

Then, the time difference of the Lyapunov function is given by

$$\begin{aligned}\Delta L_C(k)&=L_C(k+1)-L_C(k)\\&=\Delta e_c(k+d)\cdot\left[2e_c(k+d)+\Delta e_c(k+d)\right]\end{aligned} \quad (22)$$

where, from (18), we have

$$\begin{aligned}\Delta e_c(k+d)&=\left(\frac{\partial e_c(k+d)}{\partial W(k)}\right)^T\Delta \boldsymbol{W}_c(k)\\&=\eta_c\left(\frac{\partial(r(k+d)-\hat{y}(k+d))}{\partial \boldsymbol{W}_c(k)}\right)^T\left(e_c(k+d)\right)\frac{\partial \hat{y}(k+d)}{\partial u(k)}\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}\\&=-\eta_c e_c(k+d)\left(\frac{\partial \hat{y}(k+d)}{\partial u(k)}\right)^2\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}^T\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}\\&=-\eta_c e_c(k+d)\left(\frac{\partial \hat{y}(k+d)}{\partial u(k)}\right)^2\left\|\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}\right\|_2^2\end{aligned} \quad (23)$$

and

$$\left\|\left(\frac{\partial u(k)}{\partial \mathbf{W}_c(k)}\right)\right\|_2^2 = \sum_{i=1}^{n}\sum_{k=1}^{K_i}\left(\left(\frac{\partial u(k)}{\partial w_{jkl}^i}\right)^2 + \left(\frac{\partial u(k)}{\partial b_{jk}^i}\right)^2\right) + \sum_{j=1}^{m}\left(\left(\frac{\partial u(k)}{\partial b_j}\right)^2 + \left(\frac{\partial u(k)}{\partial w_j}\right)^2\right) + \sum_{i=1}^{n}\sum_{k=1}^{K_i}\left(\left(\frac{\partial u(k)}{\partial w_k^i}\right)^2 + \left(\frac{\partial u(k)}{\partial w_{jk}^i}\right)^2\right) \tag{24}$$

Thus, from (22) and (23), it follows that

$$\Delta L_C(k) = -\eta_c e_c^2(k)\left\|\frac{\partial \hat{y}(k)}{\partial \boldsymbol{W}_c(k)}\right\|_2^2 \cdot \left[2 - \eta_c \left\|\frac{\partial \hat{y}(k)}{\partial \boldsymbol{W}_c(k)}\right\|_2^2\right] \tag{25}$$

Obviously, $\Delta L_C(k) < 0$ is negative definite if and only if the sufficient condition (19) is satisfied. This completes the proof.

Theorem 4: The best learning rate $\eta_c^*(k)$ for the BLS-PIDLC controller satisfies the following condition

$$\eta_c^* = \frac{1}{\max_k \left(\frac{\partial \hat{y}(k+d)}{\partial u(k)}\right)^2 \left\|\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}\right\|_2^2} \tag{26}$$

Proof: Similar to Theorem 2, the best control learning rate $\eta_c^*(k)$ can be easily found from (27) by solving $d\Delta L_C(k)/d\eta_c = 0$ in (22) at the time instant k where

$$\Delta L_C(k) = -e_c^2(k+d)\max_k \left(\frac{\partial \hat{y}(k+d)}{\partial u(k)}\right)^2 \left\|\left(\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}\right)\right\|_2^2 \cdot \left[2\eta_c - \eta_c^2 \max_k \left(\frac{\partial \hat{y}(k+d)}{\partial u(k)}\right)^2 \left\|\left(\frac{\partial u(k)}{\partial \boldsymbol{W}_c(k)}\right)\right\|_2^2\right] \tag{27}$$

This completes the proof.

Remark 2: In applying Theorem 3, it is required that the control learning rate, η_c ,must be checked at every sampling time instant, in order to guarantee that the inequality (19) holds. Obviously, the least upper bound, $\max\left\|\partial u(k)/\partial \mathbf{W}_c(k)\right\|$, is large when the learning phase just begins, but the bound will become small and even zero once the learning phase has been finished. Hence, it is a safe way to select a small initial learning rate of η_c to achieve the uniform asymptotical stability of the proposed BLS-PIDLC closed-loop control system.

3. Iterative BLS-APIDLC Algorithm

In what follows, an iterative BLS-APIDLC control algorithm is proposed to on-line update not only the parameters of the BLS identifiers, but also the parameters of the proposed predictive BLS-PIDLC controller, thus achieving desired control iteratively. The computations at each time instant k can be described as follows;

Step 1. Determine d, n_u and n_y of the controlled plant, and select the structure of the used BLS identifier.

Step 2. Determine the initial learning rates, η_I and η_C , and randomly initialize six type of parameters of the BLS identifier and BLS-PIDLC.

Step 3. Measure the plant output $\boldsymbol{y(k)}$ and read the reference setpoint $r(k)$.

Step 4. Compute the output $u(k)$ of the BLS-PIDLC in (16).

Step 5. Output the control signal $u(k)$ to the controlled plant in (4).

Step 6. Update all the parameters of the BLS identifier using (7) and the best learning rate $\eta_I^*(k)$ in (14).

Step 7. Update all the parameters of the BLS-PIDLC parameters using (18) and the best learning rate $\eta_c^*(k)$ in (26).

Step 8. Repeat Steps 3-7.

The following theorem summarizes the main results regarding the overall BLS-APIDLC system.

Theorem 5: Under the proposed iterative BLS-APIDLC algorithm, the overall BLS-APIDLC system is uniformly asymptotically stable if both sufficient conditions (8) and (19) hold.

Proof: Theorem 5 can be easily proven by selecting the discrete-time Lyapunov function, $L(k)$, as the sum of $L_I(k)$ and $L_C(k)$, that is, $L(k) = L_I(k) + L_C(k)$, and showing that the time difference of $L(k)$ is negative definite if both sufficient conditions (8) and (19) are satisfied.

V. SIMULATIONS AND DISCUSSION

In this section, two illustrative examples are given to explore the effectiveness and superiority of our proposed BLS-APIDLC controller by controlling some nonlinear discrete-time dynamic systems. The topology of the proposed BLS is kept fixed in our simulations. There are many different performance criterions that are used to evaluate the entire dynamic responses for controller design. Our evaluation method concentrates on the following commonly used control criteria: maximum errors, RMSE, ISE, IAE and ITAE. Performance comparisons are executed to indicate the advantageous capabilities of the proposed control method in comparison with four existing adaptive PID controllers: RBFNN-APPID (Tung, 2012), ORFWNN-APPID (Tsai et al., 2019), FWNN-APPID (Tsai et al., 2017) and FBLS-APPID where the FBLS is described in (Feng and Chen, 2018). Note that, in BLS-APPID, the BLS is employed to on-line tune the PID gains (Li, 2019).

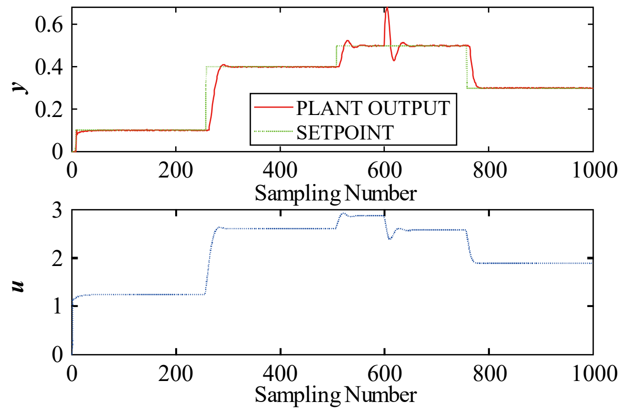

Fig. 4. Tracking responses and control signals corresponding to the large step disturbance with a peak amplitude of 0.1.

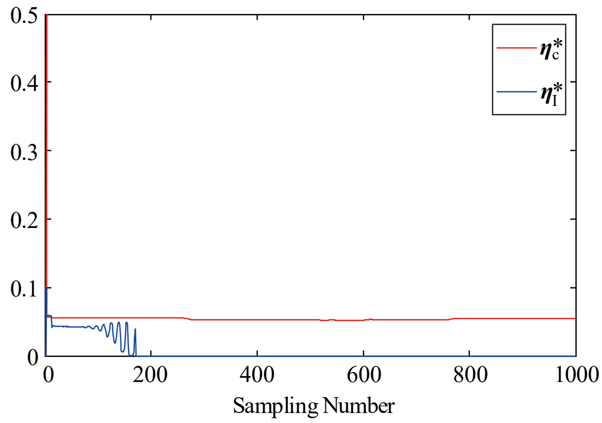

Fig. 5. Time evolutions of the learning rates of η_I^* and η_c^* for Example 1.

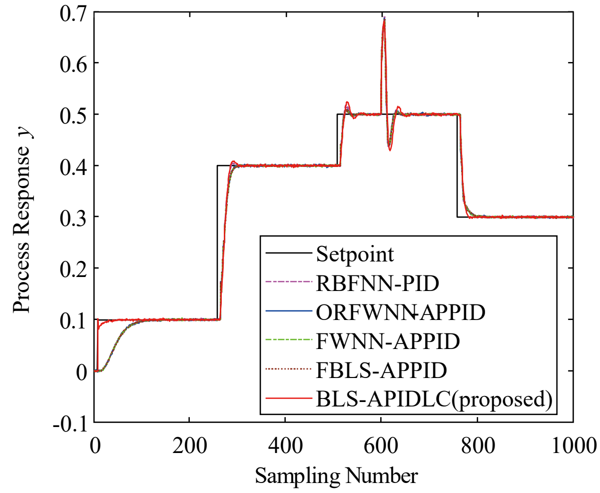

Fig. 6. Comparison of the output responses.

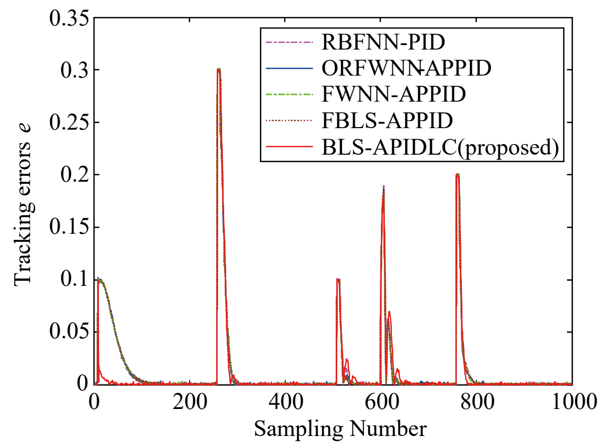

Fig. 7. Comparison of the tracking errors.

Example 1: Consider the following nonlinear discrete-time dynamic system, which is a modified version of the example (Tsai et al., 2017; Tsai et al., 2019):

$$y(k) = y^3(k-1) - 0.2|y(k-1)|u(k-d) + 0.08u^2(k-d) + \xi(k)$$

where the time delay is set by $d = 7$ and $\xi(k)$ denotes the exogenous disturbance. The design parameters are specified as $n_y = 1$, $n_u = 7$, $\eta_I = 0.1$, η_c is initially set as 0.5, and the sampling epoch S = 1000. In the example, the parameter settings of the used BLS is given as follows; the number of its inputs is selected as 3, the number of the mapping feature nodes is 4, the number of the mapping feature window is 1, and the number of the enhancement nodes is 4.

After finishing the simulation, Fig. 4 depicts the tracking responses of the proposed controller for the case of the step disturbance with a peak amplitude of 0.1 after the 600-th sampling instant. The curves of the learning rates η_I^* and η_c^* are plotted in Fig. 5. Fig. 6 compares the output responses of the BLS-APIDLC, BFNN-PID, RFWNN-APPID, FWNN-APPID and FBLS-APPID under set-point varies and large external disturbances.

Fig. 7 illustrates the tracking errors of the BLS-APIDLC, BFNN-PID, RFWNN-APPID, FWNN-APPID, FBLS-APPID and BLS-APIDLC. In addition, the results in Table 1 indicate that the four performance indexes of the proposed BLS-APIDLC controller are less than those of four existing controllers, thus indicating that the proposed controller exhibits better disturbance rejection and control performance in terms of maximum errors, RMSE, ISE and IAE. The comparative results in Table 2 illustrate that the proposed BLS-APIDLC with different initial rates of η_I and η_c has the almost robustness property of the proposed BLS-APIDLC in terms of maximum errors, RMSE, ISE and IAE.

Example 2: Consider the following nonlinear discrete-time dynamic system, which is a modified version of the example (Tsai et al., 2017; Tsai et al., 2019):

Table 1. Comparison of performance indexes of the comparatuve controllers

Controller	Max. Error	RMSE	ISE	IAE
RBFNN-PID (Tung, 2012)	0.3015	0.0442	1.9498	14.5264
ORFWNN-APPID (Tsai et al., 2019)	0.3010	0.0442	1.9580	14.6306
FWNN-APPID (Tsai et al., 2017)	0.3026	0.0443	1.9616	14.6509
FBLS-APPID (Feng and Chen, 2018)	0.3010	0.0442	1.9578	14.6338
BLS-APIDLC	0.3001	0.0419	1.7534	11.3588

Table 2. Comparison of performance indexes of different initial learning rates for BLS-PIDLC in Example 1.

(η_I, η_c)	Max. Error	RMSE	ISE	IAE
(0.25,0.25)	0.3011	0.0420	1.7643	11.4199
(0.5,0.5)	0.301	0.0419	1.7536	11.4485
(1,1)	0.3014	0.0419	1.7568	11.4483
(0.1,0.5)	0.3001	0.0419	1.7534	11.3588

$$\begin{aligned} y(k) &= 0.9722y(k-1) + 0.3578u(k-d) - 0.1295u(k-d-1) \\ &- 0.3103y(k-1)u(k-d) - 0.04228y^2(k-2) \\ &+ 0.1663y(k-2)u(k-d-1) - 0.03259y^2(k-1)(k-2) \\ &- 0.3515y^2(k-1)u(k-d-1) \\ &+ 0.3084y(k-1)(k-2)u(k-d-1) \\ &+ 0.1087y(k-2)u(k-d)u(k-d-1) + v(k) \end{aligned}$$

where the time delay is set by $d = 3$. The design parameters are specified as $n_y = 2$, $n_u = 3$, $\eta_I = 0.1$, η_c is initially set as 0.6, and the sampling epoch is 1000. Note that the used controller works as a P-like controller such that the number of the input number is 1, the number of the mapping feature nodes is 4, the number of the mapping feature window is 1, and the number of the enhancement nodes is 10. The control purposes are not only to let the output of the BLS approximate the system output $y(k)$ as closely as possible, but also to control the system output to track the subsequent reference $r(k)$ and reject external disturbances $v(k)$

$$r(k) = \begin{cases} 1, & 0<k \le 500 \\ 0, & 500<k \le 1000 \end{cases}, \quad v(k) = \begin{cases} 0, & 0<k \le 250 \\ 0.05, & 250<k \le 750 \\ 0.1, & 750<k \le 1000 \end{cases}$$

Fig. 8 illustrates the tracking responses and control signals of the proposed method. The curves of the learning rates η_I^* and η_c^* are given in Fig. 9. Fig. 10 shows the comparison results of the output responses for the proposed BLS-APIDLC controller and the four existing controllers under set-point changes and disturbances. Fig. 11 demonstrates the tracking

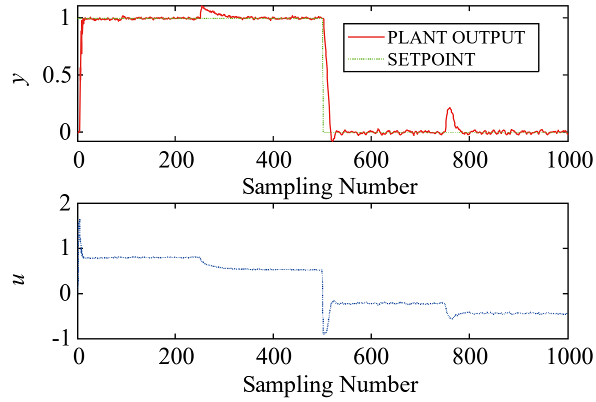

Fig. 8. Tracking responses and control signals in Example 2.

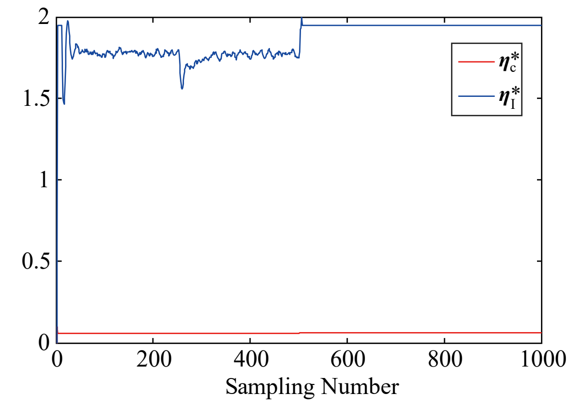

Fig. 9. Time evolutions of the learning rates of η_I^* and η_c^* for Example 2.

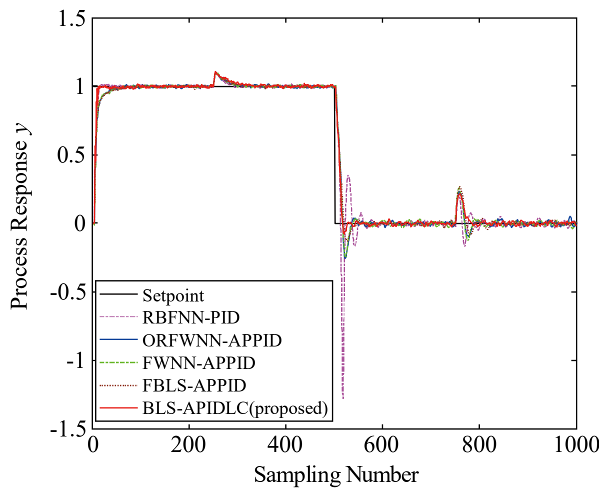

Fig. 10. Comparative output responses of the five controllers.

Table 3. Comparison of performance indexes of the comparatuve controllers

Controller	Max. Error	RMSE	ISE	IAE	ITAE
RBFNN-PID (Tung, 2012)	1.2793	0.1344	18.0695	38.4014	18242
ORFWNN-APPID (Tsai et al., 2019)	1.0029	0.1185	14.0375	34.6003	14676
FWNN-APPID (Tsai et al., 2017)	1.0030	0.1171	13.7115	33.4541	13870
FBLS-APPID (Feng and Chen, 2018)	1.0024	0.1180	13.9252	33.2663	14878
BLS-APIDLC	1	0.1102	12.1415	27.7630	11718

Table 4. Comparison of performance indexes of different initial learning rates for BLS-PIDLC in Example 2.

(η_I, η_c)	Max. Error	RMSE	ISE	IAE	ITAE
(0.25,0.25)	1.0013	0.1104	12.1808	28.4248	12242
(0.5,0.5)	1.0130	0.1120	12.5526	29.4484	12368
(1,1)	1.0040	0.1150	13.2194	30.1088	12169
(0.1,0.6)	1	0.1102	12.1415	27.7630	11718

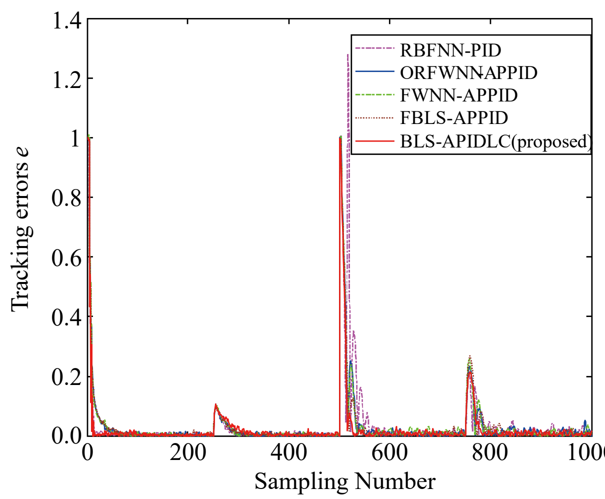

Fig. 11. Comparative tracking errors of the five controllers.

errors of these five controllers. The comparative results in Table 3 reveal that the proposed BLS-APIDLC outperforms the four controllers in terms of maximum errors, RMSE, ISE, IAE and ITAE. The results in Table 4 show that the proposed BLS-APIDLC with different initial rates of η_I and η_c exhibits the almost robustness property of the proposed BLS-APIDLC in terms of maximum errors, RMSE, ISE, IAE and ITAE.

V. CONCLUSIONS

This paper has proposed an adaptive PID-like controller (PIDLC) using broad learning system (BLS) for unknown nonlinear discrete-time dynamic systems with time delays. This special controller, abbreviated BLS-APIDLC, makes the BLS network able to work as a PID controller as needed. The BLS is employed to successfully learn the incremental characteristics of the digital nonlinear dynamic systems and the BLS-PIDLC is effectively designed for setpoint tracking disturbance rejection of such unknown nonlinear dynamic systems. In the outset, two sets of iterative gradient descent learning algorithms for the BLS-APIDLC method have been established, and its overall closed-loop asymptotical stability has also been investigated to find two sufficient conditions regarding the learning rates. The effectiveness and superiority of the proposed adaptive BLS-APIDLC method have been well exemplified by conducting two simulations on widely known digital nonlinear time-delay dynamic systems. An interesting future work would be to propose a BLS-APIDLC control method for unknown nonlinear multi-input-multi-output (MIMO) discrete-time dynamic systems.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support from the Ministry of Science and Technology (MOST), Taiwan, the Republic of China, under contract MOST 108-2634-F-005-002-

REFERENCES

Andò, B., S. Graziani and M. G. Xibilia (2018). Low-order nonlinear finite-impulse response soft sensors for ionic electroactive actuators based on deep learning. IEEE Trans. Instrumentation and Measurement 68 (5), 1637-1646.

Astrom, K. and T. Hagglund (1995). PID Controllers: Theory, Design, and Tuning, 2nd Edition. Instrument Society of America.

Chen, C. L. P. and Z. L. Liu (2018). Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. IEEE Transactions on Neural Networks and Learning Systems 29 (1), 10-24.

Chen, C. L. P., Z. L. Liu and S. Feng (2019). Universal approximation capability of broad learning system and its structural variations. IEEE Transactions on Neural Networks and Learning Systems 30 (4), 1191-1204.

Cong, S. and Y. Liang (2009). PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. IEEE Transactions on Industrial Electronics 56 (10), 3872-3879.

Du, J., C. Vong and C. L. P. Chen (2020). Novel efficient RNN and LSTM-Like architectures: recurrent and gated broad learning systems and their applications for text classification. IEEE Transactions on Cybernetics (early Access).

Fahmy, R. A., R. I. Badr and F. A. Rahman (2014). Adaptive PID controller using RLS for SISO stable and unstable systems. Advances in Power Electronics 24, 1-5.

Feng, S. and C. L. P. Chen (2018a). Fuzzy broad learning system: a novel neuro-fuzzy model for regression and classification. IEEE Transactions on Cybernetics 50 (2), 414-424.

Feng, S. and C. L. P. Chen (2018b). Broad learning system for control of nonlinear dynamic systems. Proceeding of 2018 IEEE International Conference on Systems, Man, and Cybernetics, Miyasaki, Japan, 2230-2235.

Fu Y. and T. Y. Chai (2011). Robust self-tuning PID-like control with a filter for a class of discrete time systems. Proceeding of 2011 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 6783-6787.

Ghazia, M. M., B. Yanikoglua and E. Aptoula (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235, 228-235.

Jin, J. W. and C. L. P. Chen (2018). Regularized robust broad learning system for uncertain data modeling. Neurocomputing 322, 58-69.

Kumar, V., P. Gaur and A. P. Mittal (2014). ANN based self-tuned PID like adaptive controller design for high performance PMSM position control. Expert Systems with Applications 41 (7), 7995-8002.

Kang, Y., S. Chen, X. Wang and Y. Cao (2019). Deep convolutional identifier for dynamic modeling and adaptive control of unmanned helicopter. IEEE Transactions on Neural Networks and Learning Systems 30 (2), 524-538.

Li, Y. C. (2019). Intelligent auto-tuning of PID controllers using fuzzy broad learning system for tool-grinding servo control systems. MS thesis, Department of Electrical Engineering, National Chung Hsing University.

O'Dwyer, A. (2009). Handbook of PI and PID Controller Tuning Rules, 3rd Edition. Imperial College Press, London.

Oliveira, J. V. and J. M. Lemos (2000). A comparison of some adaptive-predictive fuzzy-control strategies. IEEE Transactions on Systems, Man, and Cybernetics Part C 30(1), 138-145.

Pan, T., S. Li and W. J. Cai (2007). Lazy learning-based online identification and adaptive PID control: a case study for CSTR process. Industrial Engineering Chemistry Research 46 (2), 472-480.

Pao, Y. H., G.-H. Park and D. J. Sobajic (1994). Learning and generalization characteristics of the random vector functional-link net. Neurocomputing 6 (2), 163-180.

Rosa, E. and W. Yu (2016). Randomized algorithms for nonlinear system identification with deep learning modification. Information Sciences 364 (C), 197-212.

Silva, G. J., A. Datta and S. R. Bhattacharyya (2005). PID Controllers for Time-Delay Systems, Springer-Verlag, New York.

Tsai, C. C., C. C. Yu and C. T. Tsai (2019). Adaptive ORFWNN-based predictive PID control. International Journal of Fuzzy Systems 21(3), 1544-1559.

Tsai, C. C., F. C Tai and Y. L. Chang (2017). Adaptive predictive PID control using fuzzy wavelet neural networks for nonlinear discrete-time time-delay systems. International Journal of Fuzzy Systems 19 (6), 1718-1730.

Tsai, P. Y., H. C. Huang, S. J. Chuang, Y. J. Chen and R. C. Hwang (2005). The model reference control by adaptive PID-like fuzzy-neural controller. Proceeding of 2005 IEEE International Conference on Systems, Man and Cybernetics 1, Waikoloa, HI, USA, 239-244.

Tsai, C. C., F. C Tai and Y. C. Wang (2014). Global localization using dead-reckoning and kinect sensors for robots with omnidirectional mecanum wheels. Journal of Marine Science and Technology 22 (3), 321-330.

Tung, S. L. (2012). Design and experimentation of digital two-degree-of-freedoms temperature controllers for pet blow molding machines. M. S. Thesis, Department of Electrical Engineering, National Chung Hsing University.

Vilanova, R. and A. Visioli (2012). PID Control in the Third Millennium, Springer-Verlag, London.

Wang, J., W. Z. Gao, S. S. Gu and F. L. Wang (1997). PID-like controller using a modified neural network. International Journal of Systems Science 28 (8), 809-815.

Xu, M. L., M. Han, C. L. P. Chen and T. Qiu (2018). Recurrent broad learning systems for time series prediction. IEEE Transactions on Cybernetics 50 (4), 1405-1417.

Yang, Q., X. Zhang, M. Li and Q. Wang (2015). An adaptive predictive control strategy based on Laguerre function sequence for chopper cascade speed control system. Proceeding of 2015 Chinese Control and Decision Conference, Qingdao, China, 2270-2275.