Adaptive Predictive PID Control Using Recurrent Fuzzy Broad Learning System for Accurate Setpoint Tracking of Digital Nonlinear Time-Delay Dynamic Systems

Ali Rospawan, Ching-Chih Tsai, and Feng-Chun Tai

***Abstract*— This paper presents a novel adaptive predictive proportional-integral-derivative (PID) control using a new recurrent fuzzy broad learning system (RFBLS) for setpoint tracking control and disturbance rejection of a class of nonlinear discrete-time dynamic systems with time delay. The proposed controller, abbreviated as RFBLS-APPID, is formed from an online RFBLS identifier for online parameter tuning and identification, and an adaptive predictive RFBLS-PID control for accurate setpoint tracking and disturbance rejection. The three-term PID controller gain parameters are automatically tuned by the RFBLS identifier. The setpoint tracking of the proposed RFBLS-APPID control method is well exemplified by conducting simulations employing two well-known nonlinear digital discrete-time time-delay dynamic systems, thus showing its effectiveness and superiority.**

Keywords: PID Control, intelligent control, predictive control, recurrent fuzzy broad learning system (RFBLS)

I. INTRODUCTION

PID parameter tuning refers to adjusting the PID controller's three-term parameters, proportional, integral, and derivative gain, to achieve the system's required control performance. In industry and academia, the self-tuning of PID control has been long considered a significant parameter search problem. Many researchers have proposed several methods for self-tuning PID gains. The author in [1] and [2] proposed a self-tuning method for the PID control, and later a fuzzy wavelet neural network (FWNN) was presented by [3] which was considered as the earlier wave of neural network application for the PID control. The authors in [4]–[6] proposed an improvement of the FWNN method. Aimed to improve the FWNN learning capabilities, the authors in [7]–[9] proposed the ORWNN, ORFWNN, and RBFNN methods for online parameter tuning for the PID controllers.

To improve the learning efficiency without complex architecture, the author in [10,11] proposed a broad learning system (BLS) and later in [12] and [13] constructed a more efficient broad learning system by employing neuro-fuzzy and labeled it as fuzzy BLS (FBLS). This FBLS is a method that incorporated Takagi-Sugeno fuzzy system inside the BLS by adding the enhancement node to enhance learning capabilities. In its applications, FBLS has been used by several researchers. The authors in [14], [15] applied the FBLS to the tool-grinding servo system, and the authors in [16], [17] applied it to the wafer cleaning machine in the semiconductor industry.

Ali Rospawan, *Ching-Chih Tsai and Feng-Chun Tai are with the National Chung Hsing University, Taichung, Taiwan (e-mail: alirospawan34@gmail.com, cctsai@nchu.edu.tw, fctai@nchu.edu.tw).

Recently, an improvement of the FBLS method has been proposed in two ways. Recurrent FBLS abbreviate as RFBLS by the authors in [16] and Output Recurrent FBLS or ORFBLS by the authors in [18] who aimed to improve the RFBLS learning capabilities, this paper is proposed.

The research objective of this paper is to propose a new RFBLS system identifier and a new adaptive predictive PID controller to form an RFBLS-APPID control method by modifying the controller structure of the RFBLS method proposed in [16]. The remaining sections of the paper are structured as follows. Section II introduces the new proposed RFBLS control structure. Section III describes the new RFBLS identifier along with its updated laws and convergent proof. Section IV elaborates on the proposed control design and presents the real-time control algorithm along with the investigation of the asymptotical stability analysis of the proposed controller. Section V compares the proposed RFBLS-APPID control method through two simulation studies using two well-known digital nonlinear discrete-time time-delay dynamic systems models. Section VI draws the conclusions and presents two future research topics.

II. NEW RFBLS CONTROL STRUCTURE

By improving the control structure proposed by [16], we propose a new RFBLS control structure depicted in Fig. 1. This RFBLS structure consists of an input layer, fuzzy subsystem layer, defuzzification layer, enhancement layer, and an output layer. The detailed explaination of each layer is described as follows.

A. Input Layer

The input node u on the input layer is composed of M numbers of nodes that are connected directly to its designated fuzzy-sets nodes.

B. Fuzzy Subsystem Layer

Following the control structure in Fig. 1, the input node U_{s1} is designated to $A^i{}_{k1}$ and input node U_{s2} is designated to $A^i{}_{k2}$…and input node U_{sM} is designated to $A^i{}_{kM}$ .Thus, we have the fuzzy rules described by

$$z_{sk}^i = f_k^i\left(u_1, u_2, ..., u_M\right), \quad k = 1, 2, ..., K_i \tag{1}$$

which adopts the Takagi-Sugeno fuzzy system first-order and aims to get a better learning performance. The new proposed structure is proposed by feedbacking the output of the enhancement node system into its second epoch back to its enhancement node, and the fuzzy rules are changed as follows.

$$z_{sk}^{i} = f_{k}^{i}(u_{s1}, u_{s2}, \ldots, u_{sM}) = \sum_{t=1}^{M} \alpha_{kt}^{i} u_{st} + a_{ft} F_{si}\left(t-1\right) \quad (2)$$

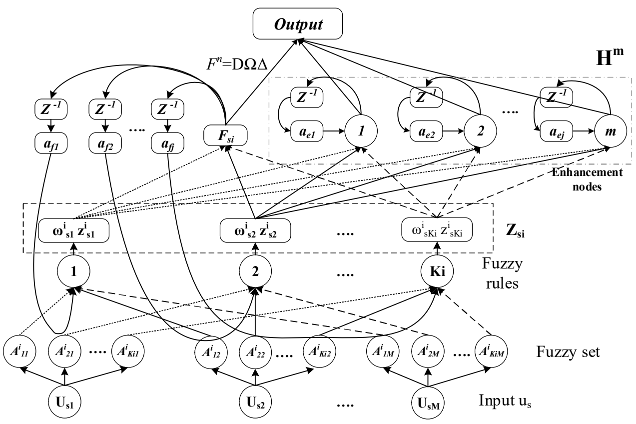

Fig.1. The structure of the proposed new RFBLS.

where α_{kt}^{i} denotes the coefficient with $0 < \alpha_{kt}^{i} < 1$ ranges of values, a_{ft} is the fuzzy subsystem weighting vector and $F_{si}(t-1)$ denotes the fuzzy subsystem output after the first iteration of the learning process. The i_{th} fuzzy subsystem for the k_{th} fuzzy rule fire strength is given by.

$$\tau_{sk}^{i} = \prod_{t=1}^{M} \mu_{kt}^{i}(u_{st}) \quad (3)$$

Then, for each fuzzy rule the weighted fire strength is described as follows;

$$\omega_{sk}^{i} = \frac{\tau_{sk}^{i}}{\sum_{k=1}^{K_i} \tau_{sk}^{i}} \quad (4)$$

Adopting the Gaussian membership function μ_{kt}^{i} in correlation to the fuzzy set A_{kt}^{i} gives

$$\mu_{kt}^{i} = \exp\left(-\frac{(u_{st} - c_{kt}^{i})^2}{\sigma_{kt}^{i\,2}}\right) \quad (5)$$

where c_{kt}^{i} and σ_{kt}^{i} are respectively the Gaussian membership function center and width.

C. Defuzzification Layer

The defuzzification layer is the output of the fuzzy subsystem which is directly linked to the output layer in line with the enhancement layer output matrix $H^{m}W_{e}$. Each fuzzy subsystem needs to be a multi-output model, as shown by the C component inside the training target $Y \in \Re^{1xC}$. The following is the i_{th} fuzzy subsystem output vector for each of u_s training samples.

$$\begin{aligned} F_{si} &= \left(\sum_{k=1}^{K_i} \omega_{sk}^{i} \left(\sum_{t=1}^{M} \delta_{k1}^{i} \alpha_{kt}^{i} u_{st} \right), \cdots, \sum_{k=1}^{K_i} \omega_{sk}^{i} \left(\sum_{t=1}^{M} \delta_{kC}^{i} \alpha_{kt}^{i} u_{st} \right) \right) \\ &= \sum_{t=1}^{M} \alpha_{kt}^{i} u_{st} \left(\omega_{s1}^{i}, \cdots, \omega_{sK_i}^{i} \right) \begin{pmatrix} \delta_{11}^{i} & \cdots & \delta_{1C}^{i} \\ \vdots & \ddots & \vdots \\ \delta_{K_i 1}^{i} & \cdots & \delta_{K_i C}^{i} \end{pmatrix} \end{aligned} \quad (6)$$

where δ_{kc}^{i} is the part parameter consequent for every fuzzy rule in the i_{th} fuzzy subsystems and $\delta_{kc}^{i}\alpha_{kt}^{i}$ (c=1, 2,…, C) is formed by the coefficient α_{kt}^{i}. For the output matrixof the i_{th} fuzzy subsystem, the top layer's aggregated output of n fuzzy subsystems becomes

$$F^{n} = \sum_{i=1}^{n} F^{i} = \sum_{i=1}^{n} D^{i}\Omega^{i}\delta^{i} \in \Re^{1\times C} \quad (7)$$

D. Enhancement Layer

This subsection declares a defuzzification output by combining the output vectors from all fuzzy rules in the i_{th} fuzzy subsystem. In addition, nonlinear activation functions are used to feed all fuzzy subsystem intermediate vectors into the enhancement node layer. As a result, the vector output of the i_{th} fuzzy subsystem in the training samples u without aggregation is indicated by

$$Z_{si} = \left(\omega_{s1}^{i} z_{s1}^{i}, \omega_{s2}^{i} z_{s2}^{i}, \cdots, \omega_{sKi}^{i} z_{sKi}^{i} \right) \quad (8)$$

with the output matrix of the i_{th} fuzzy subsystem in the form

$$Z_{i} = Z_{si} \in \Re^{1\times Ki},\ i = 1, 2, \cdots, n \quad (9)$$

To maintain the symbol consistency, the intermediate output matrix of the n fuzzy subsystems is defined as

$$Z^{n} = \left(Z_{1}, Z_{2}, \cdots, Z_{n} \right) \in \Re^{1\times(K_1 + K_2 + \ldots + K_n)} \quad (10)$$

where Z^n is fed into the enhancement layer nodes. Considering L_j neurons in the j_{th} enhancement node group, the output matrix of the enhancement-layer nodes is arranged as follows;

$$H^{m} = \left(H_{1}, H_{2}, \cdots, H_{m} \right) \in \Re^{1\times(L_1 + L_2 + \ldots + L_m)} \quad (11)$$

with $H_{j}(t) = \xi_{j}\left(Z^{n} W_{hj} + \beta_{hj} + a_{ej} H_{j}(t-1) \right) \in \Re^{1\times L_j}$ being the output in a matrix form of the j_{th} enhancement nodes group. W_{hj} is the weighting matrix linking the fuzzy subsystem, a_{ej} is the weighting vector for the enhancement node, β_{hj} is the bias term and along with the fuzzy subsystem output, Z^n forms the output matrix of each enhancement node's group output that will be directly assigned to the output layer.

E. Output Layer

The top layer or output layer is the summation of the enhancement node's weighted outputs along with all fuzzy subsystem outputs. Let the enhancement layer weight matrix denote as $W_{e} \in R^{(L_1 + L_2 + \ldots + L_M)\text{x}C}$ and the weights of the fuzzy subsystems toward top layer should be unity. At last, the final output of the FBLS is described as

$$\hat{Y} = F^{n} + H^{m}W_{e} = \sum_{i=1}^{n} D^{i}\Omega^{i}\delta^{i} + H^{m}W_{e} \quad (12)$$

III. RFBLS IDENTIFIER

The new RFBLS identifier is a system identifier that learns the input-output behavior of a class of general nonlinear digital discrete-time time-delay dynamic control system models. The RFBLS identifier is used for online learning the subsequent NARMA delayed form.

$$y(k) = f\begin{pmatrix} y(k-1), y(k-2), \cdots, y(k-n_y), \\ u(k-d), \cdots, u(k-n_u) \end{pmatrix} \quad (13)$$

where f denotes the smooth nonlinear function, the real system output y is with arrangement order of n_y, the controller output u is with arrangement order of n_u, and d represents the delay time. To develop an online iterative learning algorithm for the proposed identifier, the system model (13) can be written by following incremental control inputs described as below.

$$y(k)=g\begin{pmatrix} y(k-1),\cdots,y(k-(n_y+1)), \\ \Delta u(k-d),\cdots,\Delta u(k-n_u) \end{pmatrix} \tag{14}$$

whether the smooth nonlinear function g is obtained from modifying the function f and $\Delta=1-z^{-1}$.

A. Update Laws for the RFBLS identifier

In this subsection, in order to acquire the learning algorithms parameters, c_{kt}^i, σ_{kt}^i, δ_c^i, W_l^j, a_{ej} and a_{fj} for the RFBLS identifier, the error objective function $E(k)$ needs to be declared as follows;

$$E(k)=\frac{1}{2}e^2(k)=\frac{1}{2}\left(y(k)-\hat{y}(k)\right)^2 \tag{15}$$

For easy derivation, let $P(k+1)=\left[c_{kt}^i\ \sigma_{kt}^i\ \delta_c^i\ W_l^j\ a_{ej}\ a_{fj}\right]^T$ represent the parameter vector of the RFBLS identifier. The vector parameter P is updated iteratively using the deepest gradient descent method by (16).

$$\begin{aligned}\boldsymbol{P}(k+1)&=\boldsymbol{P}(k)-\eta\frac{\partial E(k)}{\partial \boldsymbol{P}(k)}=\boldsymbol{P}(k)+\eta\left(y(k)-\hat{y}(k)\right)\frac{\partial \hat{y}(k)}{\partial \boldsymbol{P}(k)}\\ &=\boldsymbol{P}(k)+\eta e(k)\frac{\partial \hat{y}(k)}{\partial \boldsymbol{P}(k)}\end{aligned} \tag{16}$$

with η denotes as the learning rate which must be real and positive values for the new RFBLS identifier, and then using partial differential equations we have the updating algorithm as follows;

$$\frac{\partial \hat{y}(k)}{\partial \boldsymbol{P}(k)}=\left(\frac{\partial y(k)}{\partial c_{kt}^i}\ \frac{\partial \hat{y}(k)}{\partial \sigma_{kt}^i}\ \frac{\partial y(k)}{\partial \delta_c^i}\ \frac{\partial \hat{y}(k)}{\partial W_l^j}\ \frac{\partial y(k)}{\partial a_{fj}}\ \frac{\partial \hat{y}(k)}{\partial a_{ej}}\right)^T$$

Note that all parameter elements inside the RFBLS identifier parameter vector **P** are trained by utilizing (16).

B. Convergent Proof of the RFBLS identifier

Following the Lyapunov stability theory, the asymptotically stability and convergent conditions of the proposed RFBLS identifier are summarized in the following theorem.

***Theorem 1*:** Let the RFBLS identifier vector parameter P be trained persistently by exciting the inputs using (16). If the identifier learning rate η aligns with the inequality condition in (17), then the new proposed RFBLS identifier is uniformly asymptotically convergent.

$$0<\eta<2/\max_k\left\|\frac{\partial \hat{y}(k)}{\partial \mathbf{P}(k)}\right\|_2^2 \tag{17}$$

***Proof*:** To show that The RFBLS identifier with the learning algorithm (16) and inequality condition (17) is asymptotically convergent, we choose the Lyapunov function in form $L_M(k)=e^2(k)=\left(y(k)-\hat{y}(k)\right)^2$.

***Theorem 2*:** The best identification learning rate of the RFBLS identifier complies with the condition stated in (18).

$$\eta^*=1/\max_k\left\|\frac{\partial \hat{y}(k)}{\partial \boldsymbol{P}(k)}\right\|_2^2 \tag{18}$$

***Proof*:** The best identifier learning rate η^* obtained by solving the differential equation of $d\Delta L_M(k)/d\eta=0$.

***Remark 1*:** To verify that the inequality (17) always holds when using Theorem 1, the identifier learning rate should be evaluated at each sample period. It can be observed that whenr the learning process begins, the upper bound of the $\max\left\|\partial\hat{y}(k)/\partial \boldsymbol{P}(k)\right\|$ is big, but as the learning process progresses, the bound becomes smaller and eventually becomes zero. In real-world applications, the use of a low identifier learning

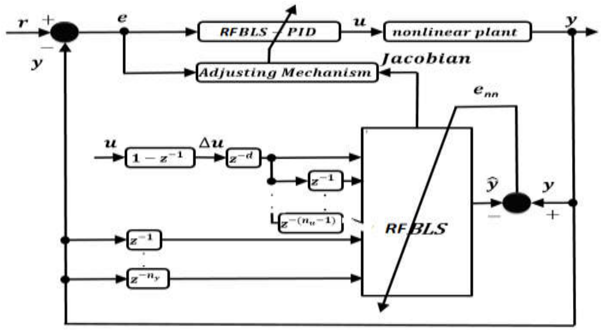

Fig. 2. The proposed RFBLS-APPID controller system.

rate is suitable and rational for ensuring the RFBLS identifier's uniform asymptotical convergence over the learning process.

IV. RFBLS-APPID CONTROL

The basic idea to online tune the RFBLS-APPID control method is to utilize the prior RFBLS identifier to do real-time tuning of the PID gains. Fig. 2 depicts the detailed description of the proposed RFBLS-APPID control system.

A. Adaptive Predictive RFBLS-PID Control

The proposed RFBLS identifier receives incremental control signals and delayed feedback outputs in the input layer and then uses the error between actual output y and estimated output $\hat{y}$ or e_{nn} to continuously update the RFBLS identifier's parameters, and then employs the Jacobian transformation to update the gain parameters of the predictive PID controller's. The velocity form of the control law is stated as follows;

$$\Delta u(k)=k_p\Delta e(k)+k_i e(k)+k_d\Delta^2 e(k) \tag{19}$$

where three-term PID parameters are denoted as k_p, k_i, and k_d. $\Delta e(k)=e(k)-e(k-1)$; $\Delta^2 e(k)=e(k)-2e(k-1)+e(k-2)$ and $\Delta u(k)=u(k)-u(k-1)$ indicates the incremental control. The tracking error is of the form $e(k)=r(k)-y(k)$ and $r(k)$ represents the designated setpoint or command.

B. Learning Algorithms for RFBLS-APPID Control Gains

This subsection describes the learning algorithm of the RFBLS-APPID controller by defining the predictive cost function relying on j-step-ahead predictive performance.

$$J(k+j)=\sum_{j=d}^{N_p}\frac{1}{2}\tilde{e}^2(k+j)=\sum_{j=d}^{N_p}\frac{1}{2}\left(y^d(k+j)-y(k+j)\right)^2 \tag{20}$$

where N_p is the predicted output horizon. In general, $\hat{y}(k+j)$ is the j-step-ahead prediction of system estimate output $\hat{y}(k)$. $r(k+j)$ means the j-step-ahead of the k-time sampling number of the given future reference signal. For easier derivations, set $\mathbf{P}_c$ to be $\mathbf{P}_c(k)=[k_p\ k_i\ k_d]^T$. Then, for every updating time $\Delta\mathbf{P}_c(k)=\mathbf{P}_c(k+1)-\mathbf{P}_c(k)=[\Delta k_p(k)\ \Delta k_i(k)\ \Delta k_d(k)]^T$ is obtained from

$$\Delta \boldsymbol{P}_c(k) = -\eta_c \frac{\partial J(k)}{\partial \boldsymbol{P}_c(k)} = -\eta_c \sum_{j=d}^{N_p} \frac{1}{2} \frac{\partial \hat{e}^2(k+j)}{\partial \boldsymbol{P}_c(k)} = \eta_c \sum_{j=d}^{N_p} \hat{e}(k+j) \frac{\partial y(k+j)}{\partial \boldsymbol{P}_c(k)} \quad (21)$$

Moreover, the partial differential form of the predictive cost function over the controller parameter vector is written as $\frac{\partial J(k)}{\partial \boldsymbol{P}_c(k)} = \left(\frac{\partial J(k)}{\partial k_p(k)} \; \frac{\partial J(k)}{\partial k_i(k)} \; \frac{\partial J(k)}{\partial k_d(k)} \right)^T$ and η_c is the control learning rate which must be real and positive. Thus, the learning algorithms in the incremental control form $\Delta \mathbf{P}_c(k) = [\Delta k_p(k) \; \Delta k_i(k) \; \Delta k_d(k)]^T$ are found by

$$\Delta k_p(k) = \eta_c \sum_{j=d}^{N_p} \hat{e}(k+j) \frac{\partial \hat{y}(k+j)}{\partial \Delta u(k)} \big(e(k) - e(k-1) \big) \quad (22)$$

$$\Delta k_i(k) = \eta_c \sum_{j=d}^{N_p} \hat{e}(k+j) \frac{\partial \hat{y}(k+j)}{\partial \Delta u(k)} e(k) \quad (23)$$

$$\Delta k_d(k) = \eta_c \sum_{j=d}^{N_p} \hat{e}(k+j) \frac{\partial \hat{y}(k+j)}{\partial \Delta u(k)} \big(e(k) - 2e(k-1) + e(k-2) \big) \quad (24)$$

To minimize the computation complexity for the predictive RFBLS-PID controller, consider the condition of $\Delta u(k+N_p) = \ldots = \Delta u(k+1) = \Delta u(k)$. Based on that speculation, the RFBLS identifier can determine system estimate output $\hat{y}(k+j)$ using the j-step-ahead predictor, and later the Jacobian transformation term in form $\partial \hat{y}(k+j)/\partial \Delta u(k)$ can be obtained.

Before closing this subsection, it is worthwhile to mention why the RFBLS identifier can improve the adaptability of the predictive PID controller. This is because once the system model has been changed and/or its parameters have been altered, the RFBLS identifier parameters will update iteratively to adjust the three-term PID parameters simultaneously to obtain satisfactory tracking performance.

C. Stability Analysis

In this subsection, we focus on the investigation of the sufficient condition for the proposed RFBLS-APPID system to be uniformly asymptotically stable. Before analyzing the controller stability, it's mandatory to assign the total of future tracking errors obtained by employing the Lyapunov function.

$$L_C(k) = \frac{1}{2} \sum_{j=d}^{N_p} (r(k+j) - \hat{y}(k+j))^2 = \frac{1}{2} \sum_{j=d}^{N_p} e^2(k+j) \quad (25)$$

This term $\hat{e}(k+j)$ is defined in (25). The time difference or incremental form of the Lyapunov function $L_C(k)$ is

$$\Delta L_C(k) = \frac{1}{2} \sum_{j=d}^{N_p} \Delta \hat{e}(k+j) \cdot \left[2\hat{e}(k+j) + \Delta \hat{e}(k+j) \right] \quad (26)$$

Theorem 3: The learning rate for the RFBLS-APPID with the convergent condition of the proposed RFBLS-APPID controller satisfies the following inequality in (27).

$$0 < \eta_c < \frac{2 \left(\sum_{m=p,i,d} \left(\sum_{j=d}^{N_p} \hat{e}(k+j) \frac{\partial \hat{y}(k+j)}{\partial k_m} \right)^2 \right)^2}{\max_k \left(\sum_{m=p,i,d} \left(\sum_{j=d}^{N_p} \left(\frac{\partial \hat{y}(k+j)}{\partial k_m} \right)^2 \right) \left(\sum_{j=d}^{N_p} \hat{e}(k+j) \frac{\partial y(k+j)}{\partial k_m} \right)^2 \right)} \quad (27)$$

Proof: Theorem 3 can be proven by assigning the Lyapunov function, $L_C(k)$, and let its time difference become negative definite, namely that $\Delta L_C(k) < 0$.

Theorem 4: Moreover, the most effective control learning rate is adopted in (28).

$$\eta_c^* = \frac{\left(\sum_{m=p,i,d} \left(\sum_{j=d}^{N_p} \hat{e}(k+j) \frac{\partial \hat{y}(k+j)}{\partial k_m} \right)^2 \right)^2}{\max_k \left(\sum_{m=p,i,d} \left(\sum_{j=d}^{N_p} \left(\frac{\partial \hat{y}(k+j)}{\partial k_m} \right)^2 \right) \left(\sum_{j=d}^{N_p} \hat{e}(k+j) \frac{\partial y(k+j)}{\partial k_m} \right)^2 \right)} \quad (28)$$

Remark 2: The upper bound of the control learning rate, η_c ,will reach infinity when $\Delta^2 e(k)$)=$e(k)$=$\Delta e(k$=0. During the start-up process, the control learning rate must be kept small. Furthermore, the Jacobian transformation of $\partial \hat{y}(k+j)/\partial \Delta u(k)$ gives a significant impact on the learning rate selection.

D. Real-Time Control Algorithm

This subsection aims to propose a real-time control and identification algorithm employing the RFBLS identifier as a system parameter identifier and an adaptive predictive PID controller. The RFBLS identifier is designed to learn the incremental dynamic model of the proposed structure model in (14). By employing the Jacobian computation and the online gain updating procedures stated in (22) - (24), the proposed RFBLS-APPID controller is well-tuned. The detailed steps are listed as below.

Step 1: Obtain and save the data of $u(k)$ and $y(k)$.
Step 2: Obtain d, n_u, and n_y for the RFBLS identifier based on the experimental input-output data.
Step 3: Assign η, K, and η_c as the positive values and real numbers, initialize k_p, k_i, and k_d from the Ziegler-Nichol PID tuning method, and initialize the size of RFBLS parameters values of δ_c^i, c_{kt}^i, σ_{kt}^i, W_l^j, a_{ej} and a_{fj}, as small random numbers.
Step 4: Evaluate $\hat{y}(k)$ in the RFBLS identifier from (12).
Step 5: Compute the PID controller control signal output $u(k) = u(k-1) + \Delta u(k)$ via (19).
Step 6: Use (16) to update the RFBLS identifier, use (17) to confirm the learning rate η, and employ the most effective identification learning rate in (18).
Step 7: Use (22)-(24) to update the PID control gains, use (27) to confirm the control learning rate η_c, and later utilize the most effective control learning rate in (28).
Step 8: Reiterate Steps 4 - 7.

The entire closed-loop system is proven to be uniformly asymptotically stable by incorporating adaptive learning of the RFBLS-APPID controller and RFBLS identifier. To produce adequate conditions, the Lyapunov function is selected from the summation of the two previously mentioned Lyapunov functions, $L_m(k)$ and $L_c(k)$, i.e., $L(k) = L_m(k) + L_C(k)$. By pursuing the proof procedures similar to the previous Sections II and III, it is simple to clarify that $\Delta L(k) = \Delta L_m(k) + \Delta L_C(k)$ should be in form of negative-definite only in the condition that the RFBLS identifier convergent condition (17) and the controller asymptotically convergent (27) can stand simultaneously.

Theorem 5: Suppose that the parameter values and assumptions of the adaptive predictive PID controller and the

RFBLS identifier are held in Theorems 1 and 3, respectively. If the RFBLS identifier convergent condition (17) and the controller asymptotically convergent (27) are both fulfilled, the entire closed-loop system equipped with the learning of RFBLS-APPID controller and RFBLS identifier is shown uniformly asymptotically stable.

***Remark 3*:** By using Theorem 5, it is important to clarify the rules of choosing the parameters in computer simulations and the results of experiments in the subsequent five guidelines. First, the RFBLS's system input order n_u and learning output order n_y are set to be more than or equal to the system input and output orders. Second, the predictive output horizon N_p is set to be more than or equal to the time delay d. If the uncertainties of the system are high, the system time delay must be smaller than N_p**.** Third, to reduce the computational load, the number of the fuzzy rules K assigned to as small as possible. Fourth, during numerical experiments or even in simulations, the learning rate of the RFBLS is reduced to meet the inequality (17) requirement. Fifth, the control learning rate η_c must satisfy the inequality (27). In real applications, the control learning rate is kept as low as possible.

V. Comparative Simulations and Discussion

In this section, the proposed new RFBLS-APPID controller's performance will be compared to the four well-known adaptive predictive PID controllers, RBFNN-PID [9], ORFWNN-APPID [8], FWNN-APPID [6], FBLS-APPID [14], [16], and as well the predecessor of the RFBLS-APPID in [17,19], by exemplifying the two well-known nonlinear digital dynamic system models in Example 1 and 2 in [6].

Before doing the numerical simulations, the RFBLS parameters follows the subsequent assignment : the number M of the inputs is 5, the fuzzy set number of each input is 4, the number of the fuzzy rules is K=4, and the number of the enhancement node group is m=4 with the sampling or epoch number being k=1000.

Example 1: Employing the nonlinear digital time-delay dynamic system model from [20], [21], the simulation system model is governed by

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

where $\varepsilon(k)$ is Gaussian white noise, $v(k)$ denotes the load disturbance, and the delay time d is assigned by 7. The desired setpoints $r(k)$ and given load disturbances $v(k)$ are shown below.

$$r(k) = \begin{cases} 0, & 0 < k \le d \\ 0.1, & d < k \le 250 \\ 0.4, & 250 < k \le 500 \\ 0.5, & 500 < k \le 750 \\ 0.3, & 750 < k \le 1000 \end{cases}, \quad v(k) = \begin{cases} 0, & 0 < k \le 600 \\ 0.1, & 600 < k \le 1000 \end{cases}$$

The following parameter settings for comparison simulations are $n_y = 2,\ n_u = d + 2,\ \eta = 0.6,\ N_p = 7$ and $\eta_c = 0.1$. The initial three-term PID gain parameters are assigned followed by $k_p = 0.6,\ k_i = 0.3$ and $k_d = 0.1$.

Example 2: Employing the nonlinear digital time-delay dynamic system model from [20] and [21], the simulation system model is governed by

$$\begin{aligned} y(k) &= 0.9722y(k-1) + 0.3578u(k-d) - 0.1295u(k-d-1) \\ &-0.3103y(k-1)u(k-d) - 0.04228y^2(k-2) \\ &+0.1663y(k-2)u(k-d-1) - 0.03259y^2(k-1)y(k-2) \\ &-0.3513y^2(k-1)u(k-d-1) + 0.3084y(k-1)y(k-2)u(k-d-1) \\ &+0.1087y(k-2)u(k-d)u(k-d-1) + \varepsilon(k) + v(k). \end{aligned}$$

TABLE I. The Controller's Performance Indexes Comparison for Simulation Example 1

Controller Name	Max Error	RMSE	ISE	IAE	ITAE
RBFNN-PID [9]	0.3026	0.0443	1.962	14.598	5224.517
ORFWNN-APPID [8]	0.3018	0.0442	1.955	14.590	5181.255
FWNN -APPID [6]	0.3024	0.0443	1.960	14.592	5218.029
FBLS-APPID [14]	0.3008	0.0442	1.953	14.548	5176.013
RFBLS-APPID [19]	0.3005	0.0442	1.950	14.540	5159.464
Proposed RFBLS-APPID	0.3008	0.0442	1.9458	14.5866	5212.033

TABLE II. The Controller's Performance Indexes Comparison for Simulation Example 2

Controller Name	Max Error	RMSE	ISE	IAE	ITAE
RBFNN-PID [9]	1.2793	0.1344	18.0695	38.4014	18242
ORFWNN-APPID [8]	1.0029	0.1185	14.0375	34.6003	14676
FWNN -APPID [6]	1.0030	0.1171	14.0375	33.4541	13870
FBLS-APPID [14]	1.0024	0.1180	13.9252	33.2663	14878
RFBLS-APPID [19]	1.0088	0.1157	13.7115	31.7528	13600
Proposed RFBLS-APPID	1.0010	0.1152	13.2779	32.7008	13468

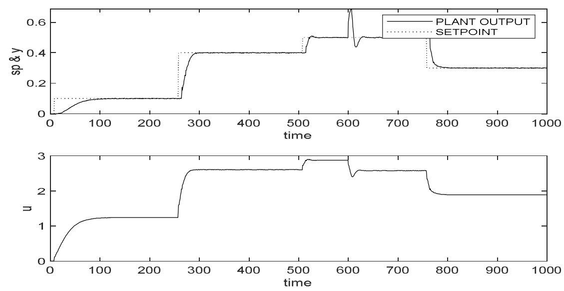

Figure 3. Setpoint tracking results and control signals of Example 1.

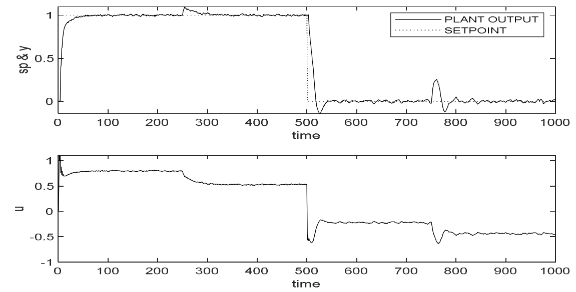

Fig. 4. Setpoint tracking results and control signals of Example 2.

where d =3, $n_y = 2, n_u = d + 2, \eta = 0.5, N_p = 3$ and $\eta_c = 0.5$. The designated setpoint $r(k)$ and load disturbance $v(k)$ are detailed as follows.

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

The initial three-term PID parameters are assigned followed by $k_p = 0.8,\ k_i = 0.1$ and $k_d = 0.2$. After conducting computer simulations for both examples, the comparative results in Table 1 and Table 2 prove the superiority of the new proposed RFBLS-APPID controller compared with existing adaptive predictive PID controller methods in terms of the maximum error, root means square error (RMSE), integral absolute error (IAE), integral of square error (ISE) and integral of time-weighted absolute error (ITAE). Note that the smaller value is better. Moreover, Fig. 3 and Fig. 4 sequentially show the setpoint tracking control results and controller signals for simulation results using Example 1 and Example 2. The results show that the proposed method is well managed to adapt and tune the PID gain parameters to follow the set point even under Gaussian white noise and load disturbances.

VI. CONCLUSIONS AND FUTURE WORK

This paper has presented a new RFBLS-APPID control method for a class of single-input-single-output nonlinear digital time-delay dynamic systems. The proposed RFBLS-APPID controller has been composed of an RFBLS identifier for system identification, and an adaptive predictive RFBLS-PID control for accurate setpoint tracking and disturbance rejection. Through comparative simulations, the proposed RFBLS-APPID method proves its superiority in comparison with the existing adaptive predictive PID controllers. In future work, the proposed RFBLS-APPID method would be applied to control the temperature or pressure control of a semiconductor manufacturing apparatus [19] or MIMO systems [22].

ACKNOWLEDGMENT

The authors deeply acknowledge financial support from the Ministry of Science and Technology (MOST), Taiwan, ROC, under contract contracts MOST 111-2622-8-005 -005 -TE1 and MOST 109-2221-E-005 -066 -MY2.

REFERENCES

[1] P. Vega, C. Prada, and V. Aleixandre, “Self-tuning predictive PID controller,” *IEE Proc. Control Theory Appl.*, vol. 138, no. 3, pp. 303–312, May 1991, doi: 10.1049/ip-d.1991.0041.

[2] T. Yamamoto, S. Omatu, and M. Kaneda, “A design method of self-tuning PID controllers,” in *Proc.of 1994 American Control Conference - ACC ’94*, Jun. 1994, vol. 3, pp. 3263–3267 vol.3.

[3] R. H. Abiyev and O. Kaynak, “Fuzzy wavelet neural networks for identification and control of dynamic plants—a novel structure and a comparative study,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 8, pp. 3133–3140, Aug. 2008, doi: 10.1109/TIE.2008.924018.

[4] C. H. Lu, “design and application of stable predictive controller using recurrent wavelet neural networks,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3733–3742, Sep. 2009, doi: 10.1109/TIE.2009.2025714.

[5] C. C. Tsai and Y. L. Chang, “Self-tuning PID control using recurrent wavelet neural networks,” in *Proc.* of *2012 IEEE Intern. Conf. on Systems, Man, and Cybernetics (SMC)*, Oct. 2012, pp. 3111–3116. doi: 10.1109/ICSMC.2012.6378269.

[6] C. C. Tsai, F. C. Tai, Y. L. Chang, and C. T. Tsai, “Adaptive predictive PID control using fuzzy wavelet neural networks for nonlinear discrete-time time-delay systems,” *Int. J. Fuzzy Syst.*, vol. 19, no. 6, pp. 1718–1730, Dec. 2017, doi: 10.1007/s40815-017-0405-z.

[7] C. C. Tsai, R. S. Liu, and F. C. Tai, “Intelligent predictive PID temperature control using ORWNN for transfer mold heating process in semiconductor die packaging machines,” in *Proc. of 2017 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, Taipei, Taiwan, Sep. 2017.

[8] C. C. Tsai, C. C. Yu, and C. T. Tsai, “Adaptive ORFWNN-based predictive PID control,” *Int. J. Fuzzy Syst.*, vol. 21, no. 5, pp. 1544–1559, Jul. 2019, doi: 10.1007/s40815-019-00650-w.

[9] C. C. Tsai, Y. L. Chang, and S. L. Tung, “Two DOF temperature control using RBFNN for stretch PET blow molding machines,” in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC2014)*, Oct. 2014, pp. 1759–1764.

[10] C. L. P. Chen and Z. Liu, “Broad learning system: an effective and efficient incremental learning system without the need for deep architecture,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[11] S. Feng and C. L. P. Chen, “Broad learning system for control of nonlinear dynamic systems,” in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2018, pp. 2230–2235.

[12] S. Feng and C. L. P. Chen, “Fuzzy broad learning system: a novel neuro-fuzzy model for regression and classification,” *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 414–424, Feb. 2020.

[13] H. S. Chen, C. C. Tsai, and F. C. Tai, “Adaptive model predictive control using iterative fuzzy broad learning system for nonlinear digital time-delay dynamics system,” in *Proc. of 2020 Intern. Conf. Fuzzy Theory and Its Applications (iFuzzy),* Hsinchu, Taiwan, Nov. 2020.

[14] C.C. Tsai, C.-C. Chan, Y.-C. Li, and F.-C. Tai, “Intelligent adaptive PID control using fuzzy broad learning system: an application to tool-grinding servo control systems,” *Int. J. Fuzzy Syst.*, vol. 22, no. 7, pp. 2149–2162, Oct. 2020, doi: 10.1007/s40815-020-00913-x.

[15] Y. C. Li, *Intelligent auto-tuning of PID controllers using fuzzy broad learning system for tool-grinding servo control systems*, Master Thesis, Department of Electrical Engineering, National Chung Hsing University, 2019.

[16] C. Y. Chou, *Intelligent adaptive PID temperature control using recurrent fuzzy broad learning systems: an application to chemical heating process in a wafer cleaning machine*, Master Thesis, Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, 2020.

[17] C. Y. Chou, C. C. Tsai, and H. S. Chen, “Intelligent adaptive PID temperature control using output recurrent fuzzy broad learning system: an application to chemical heating process in a wafer cleaning machine,” in *Proc. of 2020 National Symposium on System Science and Engineering*, National Chung Hsing University, Taichung, May 2020.

[18] A. Rospawan, C. C. Tsai, and F. C. Tai, “Intelligent PID temperature control using output recurrent fuzzy broad learning system for nonlinear time-delay dynamic systems,” in *Proc. of 2022 Intern. Conf.*

on System Science and Engineering, National Chung Hsing University, Taichung, Taiwan, May 2022.

[19] C. H. Yang, C. C. Tsai, and F. C. Tai, "Adaptive nonlinear PID control using RFBLS for digital nonlinear dynamic systems," in Proc. of 2021 International Automatic Control Conference (CACS), National Chung Cheng University, Chiayi, Taiwan, Nov. 2021.

[20] C. C. Tsai and Y. R. Cheng, "Intelligent PID injection speed and pressure control using ORBLS for hydraulic plunger machine in semiconductor die packaging," the *Proc. of 2020 International Conference on System Science and Engineering*, Sep. 2020.

[21] G. S. Hung and C. C. Tsai, "Adaptive nonlinear PID control using output recurrent broad learning system for discrete-time nonlinear dynamic systems," in *Proc. of 2021 International Conference on System Science and Engineering (ICSSE)*, Aug. 2021, pp. 482–489.

[22] C. C. Tsai, G. L. Liou, and F. C. Tai, "Adaptive nonlinear tracking control using output recurrent fuzzy broad learning system for digital nonlinear MIMO dynamic systems," in Proc. *of the 2021 International Automatic Control Conference (CACS)*, National Chung Cheng University, Chiayi, Taiwan, Nov. 2021.