IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 January 2024, accepted 21 January 2024, date of publication 29 January 2024, date of current version 8 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359293

RESEARCH ARTICLE

Output Recurrent Fuzzy Broad Learning Systems for Adaptive MIMO PID Control: Theory, Simulations, and Application

ALI ROSPAWAN, CHING-CHIH TSAI, (Fellow, IEEE), AND CHI-CHIH HUNG
Department of Electrical Engineering, National Chung Hsing University, Taichung 40227, Taiwan

Corresponding author: Ching-Chih Tsai (cctsai@nchu.edu.tw)

This work was supported in part by the National Science and Technology Council (NSTC), Taiwan, under Contract NSTC 112-2221-E-005-077-.

ABSTRACT This paper proposes a novel adaptive predictive Proportional-Integral-Derivative (PID) controller utilizing an output recurrent fuzzy broad learning systems (ORFBLS) for Multiple-Input Multiple-Output (MIMO) digital control systems, aiming to effectively adapt to changing setpoints and dynamic environments. The proposed controller, MIMO ORFBLS-APPID controller in short, is proposed to extend the application of ORFBLS as an adaptive adjustment mechanism for PID gains parameters, where the controller gain matrices are automatically tuned over time by employing the Jacobian transformations of the MIMO ORFBLS identifier. Three theorems are established to ensure proper usage and successful applications of the proposed controller. The setpoints tracking control performance and disturbance rejection abilities are firmly illustrated by performing three simulations to the multivariable nonlinear dynamic systems. Moreover, one experimental study to the laboratory-built extrusion barrel in a plastic injection molding machine is done to validate the effectiveness and practicality of the proposed control method. Through comparative simulations and experimental results, the proposed controller has been shown to outperform two existing control methods in terms of control performance indexes.

INDEX TERMS Adaptive control, auto-tuning, intelligent control, output recurrent fuzzy broad learning systems (ORFBLS), parameter adjustment algorithm, PID controller.

I. INTRODUCTION

Adaptive control methods offer strong solutions for nonlinear industrial processes, while the conventional fixed gain proportional-integral-derivative (PID) controller offers a simple yet approachable control. The well-established PID mechanism compares setpoint trajectories and measured outputs for continuous corrections based on error-dependent gains. Tuning these proportional, integral, and derivative terms online and in real-time enables tailored responses adapting to changing setpoints, disturbances, or deviations [1], [2], [3], [4], [5].

To achieve an adaptive predictive PID control approach, there are several recent studies that have influenced the writing of this paper. The study in [6] proposed an adaptive PID controller by leveraging the gains dynamics of discrete PID with an adaptive time-delay control (TDC). The equivalence relationship between TDC and PID controller is used to derive the adaptive gains. The authors in [7] proposed an adaptive PID approach by utilizing adaptive fuzzy gain-scheduling (AFGS) and applied it for temperature control regulation of a continuous stirred tank reactor (CSTR), in which the adaptive fuzzy logic controller effectively adjusted the gains and compensates CSTR model nonlinearity, which evolved around its dynamic operating trajectory.

On the other hand, the following studies introduced the use of fuzzy as an adaptive control and system identification that gave another great insight into how the fuzzy system works. The study in [8] proposed a performance

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiguang Feng.

improvement of induction motor speed response by replacing the proportional-integral (PI) controller with an adaptive neuro-fuzzy inference system (ANFIS) controller. Another study [9] introduced dynamic surface control into the adaptive backstepping recursive design algorithm along with the Nussbaum function technique to tackle unknown control directions and fuzzy logic systems to identify unknown nonlinear functions. Similarly, the authors in [10], [11], and [12] integrated dynamic surface control and adaptive backstepping algorithms by utilizing fuzzy logic to identify unknown nonlinear functions and the Nussbaum function method to tackle unknown control directions. Finally, all those previously mentioned studies provided valuable insights into how adaptive algorithms perform in different applications. Specifically, adapting fuzzy logic systems to identify unknown nonlinear functions in each step of the recursive learning processes can enhance system identification. Further enhancements of fuzzy logic to identify unknown nonlinear functions for better system identification might be possible by enhancing the fuzzy structure into a fuzzy broad learning system (FBLS).

FBLS is a kind of flat two-layered neural network presented by integrating the Takagi-Sugeno (TS) fuzzy subsystem [13] into a broad learning system (BLS) [14], [15]. This FBLS assigns each of the fuzzy set membership values or the intermediate outputs of the TS fuzzy subsystem before aggregating it into the output layer to produce the fuzzy system output, as the input to each enhancement layer node inside BLS, and then combines the TS fuzzy subsystem outputs with all the enhancement layer outputs along with their weights to generate the FBLS output [14], [16], [17], [18].

Initially, employing FBLS as the compensator has been proposed by the authors in [19] and [20], where the FBLS has been proposed to recursively update the three-term gains of PID control and utilized it to control the tools grinding servo control systems with a robust and better result compared to the conventional control method. Tending to improve the effectiveness of the FBLS-based adaptive PID control, many researchers have proposed structural learning improvements of FBLS that have led to better results. The studies in [21], [22], and [23] proposed a recurrent FBLS method for adaptive predictive PID control that gave a significant improvement over the controller's performance and robustness. This aforementioned method, abbreviated as RFBLS-APPID in short, has been done by feeding back the fuzzy subsystem output back to the fuzzy rules. On the other hand, the studies in [24], [25], [26], and [27] proposed another structural improvement with output recurrent FBLS which has been done by feeding back the FBLS system output back to the fuzzy rules and back to each node in the enhancement layer. Such a method was shown to give better controller adaptability over the nonlinear system compared with the RFBLS-APPID controller, yet both methods proved applicable and effective in handling nonlinear processes. Prior to the previously mentioned studies, all mentioned control methods were applied to the single-input single-output (SISO) control systems, and they were much easier to be designed and synthesized by comparing to multi-input multi-output (MIMO) systems.

Motivated by [24], [25], [26], and [27], the objectives of this paper are to theoretically present a novel MIMO ORFBLS-APPID controller by first constructing a MIMO ORFBLS identifier to online learn the input-output dynamic behavior generated by a class of nonlinear digital dynamic systems, and then automatically tuning PID gain matrices over time by employing the Jacobian transformations of the MIMO ORFBLS identifier, and to validate the proposed MIMO ORFBLS-APPID controller via three comparative simulations and one experimental study on a real MIMO extrusion barrel. By comparing to existing methods and the state of the art, the main contributions of the paper are summarized as follows. First, a stable adaptive MIMO ORFBLS-APPID controller is theoretically presented by using the discrete-time Lyapunov stability theory. Second, sufficient conditions are found for ensuring proper usage and successful applications of such a controller. Third, a real-time adaptive control algorithm is proposed for software implementation of this type of controller.

Furthering the application of FBLS as a system compensator and identifier for an adaptive predictive PID control in the MIMO system is proposed in this study with the details being explored in the following sections. Section II provides ORFBLS system structures in detail, and Section III formulates the multivariate identifier topology. Section IV covers integrating this structure into MIMO PID gain projection. Section V proceeds with the closed-loop stability analysis of the overall control systems along with the real-time control algorithm of the proposed MIMO ORFBLS-APPID. Section VI carries out three numerical simulations to evaluate the controller performance while Section VII conducts the experimental validation to an extrusion barrel in an injection molding machine and compares the results from this experimental study with the simulation result obtained from Example 3. Lastly, Section VIII draws conclusions and presents future work.

II. OUTPUT RECURRENT FUZZY BROAD LEARNING SYSTEMS

The MIMO ORFBLS is a structural improvement of the FBLS proposed in [14] and [28] to work under MIMO conditions. From **Fig. 1,** it can be seen that compared to the FBLS structure in [14] and [28], the new structure has two improvements. The first improvement is an addition of delayed recurrent feedback from the output layer to the fuzzy rules and the second improvement is an addition of delayed recurrent feedback from the output layer back to enhancement nodes [29], [30].

In this paper, we aim to implement the MIMO ORFBLS as a lightweight controller compatible with the STM32-F746ZG microcontroller, where in this simplified version, there is only one group of enhancement layers with m number of enhancement nodes [31]. Then, by looking at **Fig. 1**, the input nodes M of input vector $\boldsymbol{x}$ in the input layer serve as external

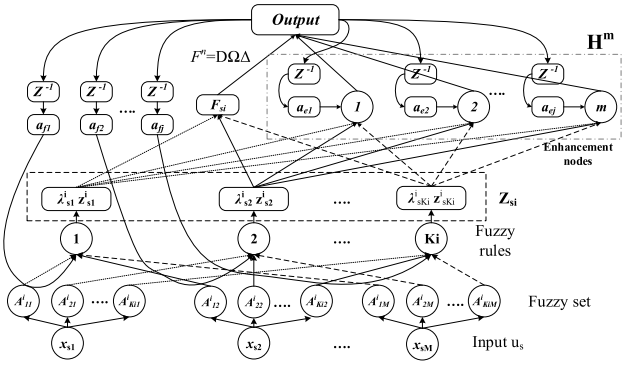

FIGURE 1. The architecture of the proposed ORFBLS.

data sources before normalizing it as fuzzy logic input, which is mathematically written by

$$\boldsymbol{x}_s(k) = (x_{s1}(k), x_{s2}(k), \cdots, x_{sM}(k)) \tag{1}$$

Then, M nodes of the input vector are normalized to be K_i number of fuzzy sets described as follows;

$$z_k^i = f_k^i\left(x_{s1}, x_{s2}, \cdots, x_{sM}\right), k = 1, 2, \ldots, K_i \tag{2}$$

Incorporating the first-order Takagi-Sugeno fuzzy neural systems with the addition of structural learning improvements, the fuzzy rules are adapted as follows;

$$z_{sk}^i = f_k^i(\boldsymbol{x}_s) = \sum\nolimits_{t=1}^{M} \alpha_{kt}^i x_{st} + p\left(\hat{y}_t\left(k-1\right) a_{ft}\right) \tag{3}$$

where the coefficient α_{kt}^i value range at [0,1], $\boldsymbol{A}_f = \left(a_{f1}, \cdots, a_{ft}\right) \in \mathbb{R}^{1\times\sum_{i=1}^{n} K_i}$ is the weight of the system estimate prior output $\hat{y}(k-1)$ and p denotes the compensator with the value range of [0, 0.5] at sampling instants k. The i_{th} fuzzy subsystem's k_{th} fuzzy rule firing strength is delineated by the following:

$$\tau_{sk}^i = \prod_{t=1}^{M} \mu_{kt}^i\left(x_{st}\right) \tag{4}$$

then, the weighted fire strength that represents the applicability degree of the fuzzy rule is described in the subsequent expression:

$$\lambda_{sk}^i = \tau_{sk}^i \Big/ \sum\nolimits_{k=1}^{K_i} \tau_{sk}^i \tag{5}$$

The fuzzy logic membership value to model the fuzzy set A_{kt}^i is detailed by

$$\mu_{kt}^i\left(x\right) = \exp\left(-\left(x - c_{kt}^i \Big/ \sigma_{kt}^i\right)^2\right) \tag{6}$$

where σ_{kt}^i and c_{kt}^i respectively denote the width and center of the Gaussian function, thus forming the degree of fuzzy set elements μ_{kt}^i at any x point.

The fuzzy subsystem intermediate output vector $\boldsymbol{Z}_{si}$ before merging it and delivering it to the enhancement node and defuzzification layer is expressed by:

$$\boldsymbol{Z}_{si} = \left(\lambda_{s1}^i z_{s1}^i, \lambda_{s2}^i z_{s2}^i, \ldots., \lambda_{sK_i}^i z_{sK_i}^i\right) \tag{7}$$

with (7), the i-th fuzzy subsystem output matrix is obtained in the following:

$$\boldsymbol{Z}_i = (\boldsymbol{Z}_{1i}, \boldsymbol{Z}_{2i}, \cdots, \boldsymbol{Z}_{Ni})^T \in \mathbb{R}^{N\times K_i}, i = 1, 2, \cdots, n \tag{8}$$

Thus, the final output of the n number of the fuzzy neural subsystem is written as follows;

$$\boldsymbol{Z} = (\boldsymbol{Z}_1, \boldsymbol{Z}_2, \cdots, \boldsymbol{Z}_n) \in \mathbb{R}^{N\times(K_1+K_2+\cdots+K_n)} \tag{9}$$

where the fuzzy subsystem output $\mathbf{Z}$ is sent to combine with m number of enhancement nodes. Then, one rewrites the output of the enhancement layer to be:

$$\boldsymbol{H} = \xi\left(\boldsymbol{Z}\boldsymbol{W}_e + \boldsymbol{\beta}_e + \hat{\boldsymbol{Y}}\left(k-1\right)\boldsymbol{A}_e\right) \in \mathbb{R}^{N\times m} \tag{10}$$

where $\mathbf{W}_e \in \mathbb{R}^{\sum_{i=1}^{n} K_i\times m}$ and $\boldsymbol{\beta}_e \in \mathbb{R}^{N\times m}$ sequentially represent the weighting parameter and bias of the matrix $\mathbf{Z}$ connected to the corresponding enhancement nodes. Different with (3), $\boldsymbol{A}_e = \left(a_{e1}, \cdots, a_{ej}\right) \in \mathbb{R}^{1\times m}$ is the weight of the feedback output, $\hat{\boldsymbol{Y}}(k-1)$, that achieves the output recurrent at enhancement nodes.

For the training sample $\boldsymbol{x}_s$ in the i-th fuzzy subsystem, the output defuzzification is expressed by

$$\boldsymbol{F}_{si} = \sum_{t=1}^{M} z_{st}^i\left(\lambda_{s1}^i, \cdots, \lambda_{sKi}^i\right)\begin{pmatrix} w_1^i \\ \vdots \\ w_{Ki}^i \end{pmatrix} \tag{11}$$

where the weighted parameter w_k^i is introduced as the fuzzy subsystem output clause of the k-th fuzzy rule. Hence, the i-th fuzzy subsystem matrix output of the training data set $\boldsymbol{X} = (\boldsymbol{x_1}, \cdots, \boldsymbol{x_N})^T \in \mathbb{R}^{N\times M}$ is written as follows;

$$\boldsymbol{F}_i = (\boldsymbol{F}_{1i}, \boldsymbol{F}_{2i}, \cdots, \boldsymbol{F}_{Ni})^T \triangleq \boldsymbol{D}\Lambda^i\boldsymbol{w}^i \in \mathbb{R}^N \tag{12}$$

where $\Lambda^i = \begin{pmatrix} \lambda_{11}^i & \cdots & \lambda_{1K_i}^i \\ \vdots & \ddots & \vdots \\ \lambda_{N1}^i & \cdots & \lambda_{NK_i}^i \end{pmatrix}$, $\boldsymbol{w}^i = \begin{pmatrix} w_1^i \\ \vdots \\ w_{Ki}^i \end{pmatrix}$ and $\boldsymbol{D} = diag\left\{\sum_{t=1}^{M} z_{sk}^i\right\}$.

Afterward, the output of the defuzzification layer that later will be aggregated to the output layer is delineated as follows;

$$\boldsymbol{F} = \sum_{i=1}^{n}\boldsymbol{F}_i = \sum_{i=1}^{n}\boldsymbol{D}\Lambda^i\boldsymbol{w}^i = \boldsymbol{D}\left(\Lambda^1, \cdots, \Lambda^n\right)\begin{pmatrix} \boldsymbol{w}^1 \\ \vdots \\ \boldsymbol{w}^n \end{pmatrix} \triangleq \boldsymbol{D}\boldsymbol{\Lambda}\boldsymbol{W}_f \in \mathbb{R}^N \tag{13}$$

where the fuzzy firing strength matrix is denoted by $\boldsymbol{\Lambda} = \left(\Lambda^1, \cdots, \Lambda^n\right) \in \mathbb{R}^{Nx(K_1+K_2+\cdots+K_n)}$ and the weighted fuzzy rules are done by setting $\boldsymbol{W}_f = w_k^i \in \mathbb{R}^{(K_1+K_2+\cdots+K_n)}$.

Now that the system output is obtained by combining previously obtained defuzzification output $\boldsymbol{F}$ and enhancement node output $\boldsymbol{H}$. Let $\boldsymbol{W}_h = w_j \in \mathbb{R}^{mx1}$ be the weight matrix that bridges the enhancement nodes and output layer, then the FBLS output is computed in the following;

$$\hat{\boldsymbol{Y}} = \boldsymbol{F} + \boldsymbol{H}\boldsymbol{W}_h \tag{14}$$

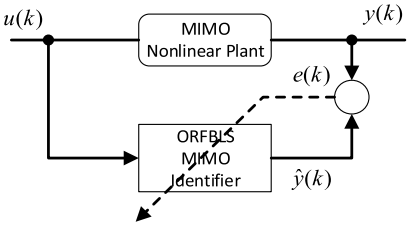

FIGURE 2. Block diagram of MIMO ORFBLS identifier.

III. MIMO ORFBLS IDENTIFIER

In this section, the MIMO ORFBLS identifier is derived to identify and update all the weighted parameters of the ORFBLS by learning the dynamics of the input-output behavior for a class of MIMO nonlinear digital discrete-time dynamic time-delayed system. To design such a system identification, the ORFBLS learning algorithm is incorporated into a nonlinear autoregressive moving-average (NARMA) model with a known time delay d in the form:

$$\begin{aligned}\boldsymbol{y}(k+1) = \boldsymbol{f}\,\big(&\boldsymbol{y}(k),\cdots,\boldsymbol{y}(k-n_y),\\ &\boldsymbol{u}(k-d),\cdots,\boldsymbol{u}(k-d-n_u)\big)\end{aligned} \tag{15}$$

where f denotes the nonlinear function vector that defines the system's input-output mapping at each time step based on its past inputs, $\boldsymbol{u}$, and outputs, $\boldsymbol{y}$. n_y and n_u are respectively the maximum number of past output and past input that are being integrated into the learning algorithm. After integrating (15) into ORFBLS, the learning process starts and holds continuously with the details presented in **Fig. 2**.

A. ORFBLS LEARNING ALGORITHM

This subsection is aimed to acquire the learning algorithm of each parameter component inside the parameter vector $\boldsymbol{P} = \left[\boldsymbol{W}_f \boldsymbol{W}_h \boldsymbol{W}_e \boldsymbol{\beta}_e \boldsymbol{A}_e \boldsymbol{A}_f\right]^T$ of the ORFBLS identifier. By following [28], [32], [33], and [34], it is necessary to find the error objective function $E(k)$ which is established from the difference among the actual system response $\mathbf{y}$ and the estimated system response $\hat{y}$ obtained from (14) in the form:

$$E(k) = \frac{1}{2}\left\|\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right\|_2^2 = \frac{\|\boldsymbol{e}\|_2^2}{2} = \frac{\boldsymbol{e}^T(k)\boldsymbol{e}(k)}{2} \tag{16}$$

The learning algorithms of the ORFBLS identifier enable the parameter vector $\boldsymbol{P}$ to be recursively updated by utilizing the deepest gradient descent approach in the following:

$$\boldsymbol{P}(k+1) = \boldsymbol{P}(k) - \eta(k)\frac{\partial E(k)}{\partial \boldsymbol{P}(k)} \tag{17}$$

where $\eta(k)$ denotes the real and positive learning rate at sampling instant k. For the sake of easier derivation, we write the parameters $\boldsymbol{P}(k)$ in a partial differential equation with respect to the error function E in the form:

$$\frac{\partial E}{\partial \boldsymbol{P}} = \left(\frac{\partial E}{\partial \boldsymbol{W}_f}, \frac{\partial E}{\partial \boldsymbol{W}_h}, \frac{\partial E}{\partial \boldsymbol{W}_e}, \frac{\partial E}{\partial \boldsymbol{\beta}_e}, \frac{\partial E}{\partial \boldsymbol{A}_e}, \frac{\partial E}{\partial \boldsymbol{A}_f}\right) \tag{18}$$

In what follows, the deepest gradient descent optimization approach is utilized to iteratively update the ORFBLS identifier's parameters. To do so, the partial derivative of each parameter is obtained by:

$$\frac{\partial E(k)}{\partial \boldsymbol{W}_f} = \boldsymbol{Z}^T(k)\left(\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right) \tag{19}$$

$$\frac{\partial E(k)}{\partial \boldsymbol{W}_h} = \boldsymbol{H}^T(k)\left(\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right) \tag{20}$$

$$\frac{\partial E(k)}{\partial \boldsymbol{W}_e} = \boldsymbol{Z}^T(k)\left(\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right)\boldsymbol{W}_h^T(k)\left(1 - \|\boldsymbol{H}\|^2(k)\right) \tag{21}$$

$$\frac{\partial E(k)}{\partial \boldsymbol{\beta}_e} = \left(\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right)\boldsymbol{W}_h^T(k)\left(1 - \|\boldsymbol{H}\|^2(k)\right) \tag{22}$$

$$\frac{\partial E(k)}{\partial \boldsymbol{A}_e} = \left(\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right)^T \boldsymbol{H}(k) \tag{23}$$

$$\frac{\partial E(k)}{\partial \boldsymbol{A}_f} = \left(\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right)^T \boldsymbol{Z}(k) \tag{24}$$

By following the iterative optimization (17), the parameters updating formula of the used ORFBLS are respectively provided by:

$$\begin{aligned}
\boldsymbol{W}_f(k+1) &= \boldsymbol{W}_f(k) - \eta(k)\frac{\partial E(k)}{\partial \boldsymbol{W}_f(k)}\\
\boldsymbol{W}_h(k+1) &= \boldsymbol{W}_h(k) - \eta(k)\frac{\partial E(k)}{\partial \boldsymbol{W}_h(k)}\\
\boldsymbol{W}_e(k+1) &= \boldsymbol{W}_e(k) - \eta(k)\frac{\partial E(k)}{\partial \boldsymbol{W}_e(k)}\\
\boldsymbol{\beta}_e(k+1) &= \boldsymbol{\beta}_e(k) - \eta(k)\frac{\partial E(k)}{\partial \boldsymbol{\beta}_e(k)}\\
\boldsymbol{A}_e(k+1) &= \boldsymbol{A}_e(k) - \eta(k)\frac{\partial E(k)}{\partial \boldsymbol{A}_e(k)}\\
\boldsymbol{A}_f(k+1) &= \boldsymbol{A}_f(k) - \eta(k)\frac{\partial E(k)}{\partial \boldsymbol{A}_f(k)}
\end{aligned} \tag{25}$$

B. CONVERGENT ANALYSIS OF MIMO ORFBLS IDENTIFIER

The uniformly asymptotically convergence condition of the MIMO-ORFBLS is established by assigning the value of learning rate η according to the discrete-time Lyapunov function in Theorem 1 [28], [32], [35], [36], [37], [38].

Theorem 1: The developed ORFBLS learning algorithm is uniformly asymptotically convergent if the identifier learning rate $\eta(k)$ complies with the following inequality condition:

$$0 < \eta(k) < 2 \Big/ \lambda_{\max}\left(\frac{\partial \hat{\boldsymbol{y}}(k)}{\partial \boldsymbol{P}}\left[\frac{\partial \hat{\boldsymbol{y}}(k)}{\partial \boldsymbol{P}}\right]^T\right) \tag{26}$$

where $\lambda_{max}(\cdot)$ denotes the maximum eigenvalues of the matrix $\frac{\partial \hat{\boldsymbol{y}}(k)}{\partial \boldsymbol{P}}\left[\frac{\partial \hat{\boldsymbol{y}}(k)}{\partial \boldsymbol{P}}\right]^T$.

Proof: Define the Lyapunov function in the form:

$$L_M(k) = \frac{1}{2}\left\|\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right\|_2^2 = \frac{1}{2}\|\boldsymbol{e}(k)\|_2^2 = \frac{\boldsymbol{e}^T(k)\boldsymbol{e}(k)}{2} \tag{27}$$

Notice that the form of the Lyapunov function between the SISO [39] and MIMO terms is different where, in MIMO, the

induced quadratic form of the error vector is used. Thus, the time difference or alteration in the Lyapunov function gives:

$$\begin{aligned}\Delta L_M(k) &= L_M(k+1) - L_M(k)\\ &= \frac{1}{2}\left(\boldsymbol{e}^T(k+1)\boldsymbol{e}(k+1) - \boldsymbol{e}^T(k)\boldsymbol{e}(k)\right)\\ &= \frac{1}{2}\left((\Delta\boldsymbol{e}(k))^T\left(2\boldsymbol{e}(k) + \Delta\boldsymbol{e}(k)\right)\right)\end{aligned} \tag{28}$$

where $\Delta\boldsymbol{e}(k) = \boldsymbol{e}(k+1) - \boldsymbol{e}(k)$ and $\boldsymbol{e}(k) = \hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)$. Furthermore, the tracking error $\Delta\boldsymbol{e}(k)$ is found as;

$$\Delta\boldsymbol{e}(k) \approx \Delta\boldsymbol{P}\left[\frac{\partial\boldsymbol{e}(k)}{\partial\boldsymbol{P}}\right]^T = \Delta\boldsymbol{P}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T \tag{29}$$

and considering the error objective function (16), the increments of the updating parameter vector $\boldsymbol{P}(k)$ are obtained from

$$\begin{aligned}\Delta\boldsymbol{P} &= -\eta(k)\frac{\partial E(k)}{\partial\boldsymbol{P}} = -\eta(k)\frac{\partial E(k)}{\partial\hat{\boldsymbol{y}}(k)}\cdot\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\\ &= -\eta(k)\left(\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)\right)\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\end{aligned} \tag{30}$$

Taking (30) as its base equation, one obtains (31) and (32).

$$\Delta\boldsymbol{e}(k) = \Delta\boldsymbol{P}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T = -\eta(k)\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T\boldsymbol{e}(k) \tag{31}$$

$$\begin{aligned}\Delta\boldsymbol{e}^T(k) &= \left(\Delta\boldsymbol{P}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T\right)^T\\ &= -\eta(k)\,\boldsymbol{e}^T\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T\end{aligned} \tag{32}$$

Finally, by substituting (31) and (32) into (28), the incremental form of the Lyapunov function $\Delta L_M(k)$ is computed in the following:

$$\begin{aligned}\Delta L_M(k) &= -\frac{1}{2}\eta(k)\boldsymbol{e}^T(k)\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T\\ &\quad\times\left(2\boldsymbol{e}(k) - \eta(k)\boldsymbol{e}(k)\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T\right)\\ &= -\frac{1}{2}\eta(k)\boldsymbol{e}^T(k)\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T\\ &\quad\times\left(2\boldsymbol{I} - \eta(k)\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T\right)\boldsymbol{e}(k)\end{aligned} \tag{33}$$

In (33), since $\boldsymbol{A} = \frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T$ be a real and symmetric matrix, so any eigenvalues of matrix $\boldsymbol{A}$ should be nonnegative and real. Next, we show that the following matrix expression is positive definite if the sufficient condition (26) is held, i.e.,

$$\boldsymbol{A}\left[2\boldsymbol{I} - \eta\boldsymbol{A}\right] > 0 \tag{34}$$

In doing so, let λ_i be one of the nonzero eigenvalues of $\boldsymbol{A}$, and $\boldsymbol{v}_i$ be the corresponding nonzero eigenvector. Thus,

$$\boldsymbol{A}\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i, \boldsymbol{v}_i \neq 0 \tag{35}$$

Next, move to find the range of the learning rate η such that

$$\boldsymbol{v}_i^T\boldsymbol{A}\left[2\boldsymbol{I} - \eta\boldsymbol{A}\right]\boldsymbol{v}_i > 0 \tag{36}$$

The property of the eigenvalue and eigenvector implies that,

$$\begin{aligned}&\boldsymbol{v}_i^T\left(2\boldsymbol{A}\boldsymbol{v}_i - \eta\boldsymbol{A}\boldsymbol{A}\boldsymbol{v}_i\right)\\ &\quad= \boldsymbol{v}_i^T\left(2\lambda_i\boldsymbol{v}_i - \eta\boldsymbol{A}\lambda_i\boldsymbol{v}_i\right)\\ &\quad= \boldsymbol{v}_i^T\left(2\lambda_i\boldsymbol{v}_i - \eta\lambda_i^2\boldsymbol{v}_i\right)\\ &\quad= \boldsymbol{v}_i^T\boldsymbol{v}_i\left(2\lambda_i - \eta\lambda_i^2\right) > 0, i = 1, \cdots, p\end{aligned} \tag{37}$$

for any i if the $2\lambda_i - \eta\lambda_i^2$ must be positive, i.e.,

$$\lambda_i\left(2 - \eta\lambda_i\right) > 0 \tag{38}$$

Considering this case, one can draw the conclusion in the form of $2-\eta\lambda_i > 0$ for any eigenvalue [39]. Hence, it follows that

$$\begin{aligned}&0 < \eta < 2/\lambda_{\max}(\boldsymbol{A})\\ &\Rightarrow 0 < \eta < \frac{2}{\lambda_{\max}\left(\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\left[\frac{\partial\hat{\boldsymbol{y}}(k)}{\partial\boldsymbol{P}}\right]^T\right)}\end{aligned} \tag{39}$$

In the end, by defining the Lyapunov function (27) and finding its time difference form is negative definite, i.e., $\Delta L_m(k) < 0$, this requires that $\lambda_i(2 - \eta\lambda_i) > 0$ for any i if the sufficient condition (26) is met. Accordingly, the proof procedure of **Theorem 1** is completed.

IV. MIMO ORFBLS-APPID CONTROLLER

This section details how the proposed MIMO ORFBLS-APPID will be established to control uncertain nonlinear MIMO dynamic systems. **Fig. 3** shows the block diagram of the proposed control system, in which the ORFBLS-APPID is done by modifying the general closed-loop PID control structure and then taking the controller output $\boldsymbol{u}(k)$ and system output $\boldsymbol{y}(k)$ as the input to the ORFBLS identifier. Later, we use the $\boldsymbol{e}_{ORFBLS}(k)$ to update the ORFBLS identifier and finally update the PID gains parameter.

A. ADAPTIVE PREDICTIVE MIMO PID CONTROLLER

In this subsection, a new Jacobian transformation derived from the ORFBLS identifier's output to the PID gains will be used for the parameter updating algorithm, so the MIMO PID control law is expressed in the following velocity form

$$\Delta\boldsymbol{u}(k) = \boldsymbol{k}_p\Delta\boldsymbol{e}(k) + \boldsymbol{k}_i\boldsymbol{e}(k) + \boldsymbol{k}_d\Delta^2\boldsymbol{e}(k) \tag{40}$$

where $\boldsymbol{k}_p$, $\boldsymbol{k}_i$ and $\boldsymbol{k}_d$ are respectively the proportional, integral, and derivative gain matrices. Further, $\boldsymbol{e}(k) = \boldsymbol{r}(k) - \boldsymbol{y}(k)$ is the vector system tracking error, and $\boldsymbol{r}(k)$ denotes the reference command or setpoints. $\Delta\boldsymbol{e}(k) = \boldsymbol{e}(k) - \boldsymbol{e}(k-1)$ and $\Delta^2\boldsymbol{e}(k) = \boldsymbol{e}(k) - 2(\boldsymbol{e}(k-1) + \boldsymbol{e}(k-2)$.

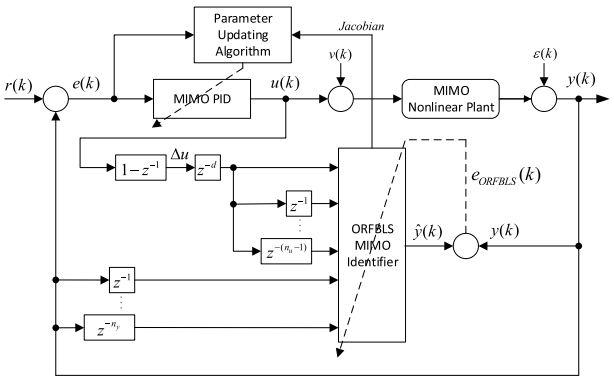

FIGURE 3. The proposed MIMO ORFBLS-APPID control system.

B. PARAMETER UPDATING ALGORITHM FOR ORFBLS-APPID

The parameter updating algorithm for the MIMO ORFBLS-APPID controller is designed to online adjust the PID gain matrices of the MIMO PID controller. This adaptation is based on the j-step-ahead predictive function performance criterion that is expressed as follows;

$$\begin{aligned}\boldsymbol{J}(k+j) &= \frac{1}{2}\sum_{j=d}^{N_p}\left\|\hat{\boldsymbol{e}}(k+j)\right\|^2 \\ &= \frac{1}{2}\sum_{j=d}^{N_p}\hat{\boldsymbol{e}}^T(k+j)\hat{\boldsymbol{e}}(k+j) \end{aligned} \tag{41}$$

where d is the system delay, $\hat{\boldsymbol{e}}(k+j) = \boldsymbol{r}(k+j) - \hat{\boldsymbol{y}}(k+j)$, and N_P is the number for the maximum predictive output horizon. Let $\boldsymbol{P}_c(k)$ be the vector containing all the three-term parameters.

$$\begin{aligned}\boldsymbol{P}_c(k) = &\begin{bmatrix} \boldsymbol{k}_{p_{11}}(k),\cdots,\boldsymbol{k}_{p_{1n}}(k),\boldsymbol{k}_{p_{n1}}(k),\cdots,\boldsymbol{k}_{p_{nn}}(k) \\ \boldsymbol{k}_{i_{11}}(k),\cdots,\boldsymbol{k}_{i_{1n}}(k),\boldsymbol{k}_{i_{n1}}(k),\cdots,\boldsymbol{k}_{i_{nn}}(k) \\ \boldsymbol{k}_{d_{11}}(k),\cdots,\boldsymbol{k}_{d_{1n}}(k),\boldsymbol{k}_{d_{n1}}(k),\cdots,\boldsymbol{k}_{d_{nn}}(k) \end{bmatrix} \\ &\in R^{3n^2\times 1}\end{aligned} \tag{42}$$

The gain update algorithm with respect to (41) is tailored to enable the vector $\boldsymbol{P}_c(k)$ to be recursively updated by using the deepest descent gradient method in the following

$$\begin{aligned}\boldsymbol{P}_c(k+1) &= \boldsymbol{P}_c(k) + \Delta\boldsymbol{P}_c(k) = \boldsymbol{P}_c(k) - \eta_c\frac{\partial \boldsymbol{J}(k)}{\partial \boldsymbol{P}_c(k)} \\ &= \boldsymbol{P}(k) - \eta_c\sum_{j=d}^{N_p}\frac{1}{2}\frac{\partial \hat{\boldsymbol{e}}^T(k+j)\hat{\boldsymbol{e}}(k+j)}{\partial \boldsymbol{P}_c(k)} \\ &= \boldsymbol{P}_c(k) + \eta_c\sum_{j=d}^{N_p}\left(\frac{\partial \hat{\boldsymbol{y}}(k+j)}{\partial \boldsymbol{P}_c(k)}\right)^T\hat{\boldsymbol{e}}(k+j)\end{aligned} \tag{43}$$

then, for every updating time instant, the incremental form $\Delta\boldsymbol{P}_c(k)$ is obtained from

$$\begin{aligned}\Delta\boldsymbol{P}_c(k) &= -\eta_c\frac{\partial \boldsymbol{J}(k)}{\partial \boldsymbol{P}_c(k)} = -\eta_c\sum_{j=d}^{N_p}\frac{1}{2}\frac{\partial \hat{\boldsymbol{e}}^T(k+j)\hat{\boldsymbol{e}}(k+j)}{\partial \boldsymbol{P}_c(k)} \\ &= \eta_c\sum_{j=d}^{N_p}\left(\frac{\partial \hat{\boldsymbol{y}}(k+j)}{\partial \boldsymbol{P}_c(k)}\right)^T\hat{\boldsymbol{e}}(k+j)\end{aligned} \tag{44}$$

where η_c is the real and positive control learning rate, and

$$\frac{\partial \boldsymbol{J}(k)}{\partial \boldsymbol{P}_c(k)} = \left(\frac{\partial \boldsymbol{J}(k)}{\partial \boldsymbol{k}_p(k)}\ \frac{\partial \boldsymbol{J}(k)}{\partial \boldsymbol{k}_i(k)}\ \frac{\partial \boldsymbol{J}(k)}{\partial \boldsymbol{k}_d(k)}\right)^T \tag{45}$$

thus, the incremental form of each component inside the element $\Delta\boldsymbol{P}_c(k)$ is respectively expressed by:

$$\Delta\boldsymbol{k}_p(k) = -\eta_c\frac{\partial \boldsymbol{J}(k)}{\partial Kp_{ij}(k)} = \eta_c\sum_{j=d}^{N_p}\left(\frac{\partial \hat{\boldsymbol{y}}(k+j)}{\partial \boldsymbol{k}_p(k)}\right)^T\hat{\boldsymbol{e}}(k+j) \tag{46}$$

where $\frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \boldsymbol{k}_p(k)} = \frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \Delta\boldsymbol{u}(k)}\frac{\partial \Delta\boldsymbol{u}(k)}{\partial \boldsymbol{k}_p(k)} = \frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \Delta\boldsymbol{u}(k)}\left(\Delta\boldsymbol{e}\left(k\right)\right)$,

$$\Delta\boldsymbol{k}_i(k) = -\eta_c\frac{\partial \boldsymbol{J}(k)}{\partial Ki_{ij}(k)} = \eta_c\sum_{j=d}^{N_p}\left(\frac{\partial \hat{\boldsymbol{y}}(k+j)}{\partial \boldsymbol{k}_i(k)}\right)^T\hat{\boldsymbol{e}}(k+j) \tag{47}$$

where $\frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \boldsymbol{k}_i(k)} = \frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \Delta\boldsymbol{u}(k)}\frac{\partial \Delta\boldsymbol{u}(k)}{\partial \boldsymbol{k}_i(k)} = \frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \Delta\boldsymbol{u}(k)}\left(\boldsymbol{e}(k)\right)$,

$$\Delta\boldsymbol{k}_d(k) = -\eta_c\frac{\partial \boldsymbol{J}(k)}{\partial Kd_{ij}(k)} = \eta_c\sum_{j=d}^{N_p}\left(\frac{\partial \hat{\boldsymbol{y}}(k+j)}{\partial \boldsymbol{k}_d(k)}\right)^T\hat{\boldsymbol{e}}(k+j) \tag{48}$$

where $\frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \boldsymbol{k}_d(k)} = \frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \Delta\boldsymbol{u}(k)}\frac{\partial \Delta\boldsymbol{u}(k)}{\partial \boldsymbol{k}_d(k)} = \frac{\partial \hat{\boldsymbol{y}}_i(k+l)}{\partial \Delta\boldsymbol{u}(k)}\left(\Delta^2\boldsymbol{e}\left(k\right)\right)$.

Finally, the PID gain parameters are updated by following

$$\boldsymbol{k}_p(k+1) = \Delta\boldsymbol{k}_p(k) + \boldsymbol{k}_p(k) \tag{49}$$

$$\boldsymbol{k}_i(k+1) = \Delta\boldsymbol{k}_i(k) + \boldsymbol{k}_i(k) \tag{50}$$

$$\boldsymbol{k}_d(k+1) = \Delta\boldsymbol{k}_d(k) + \boldsymbol{k}_d(k) \tag{51}$$

C. STABILITY ANALYSIS OF THE CONTROL SYSTEM

This subsection will target at the investigation of the sufficient learning rate η_c for the raised MIMO ORFBLS-APPID controller to be uniformly asymptotically stable. This goal can be accomplished by defining an appropriate Lyapunov function in the following manner

$$\begin{aligned}\boldsymbol{L}_C(k+j) &= \frac{1}{2}\left\|\boldsymbol{r}(k+j) - \hat{\boldsymbol{y}}(k+j)\right\|_2^2 = \frac{1}{2}\left\|\hat{\boldsymbol{e}}(k+j)\right\|_2^2 \\ &= \frac{\left\|\hat{\boldsymbol{e}}(k+j)\right\|_2^2}{2} = \frac{\hat{\boldsymbol{e}}^T(k+j)\hat{\boldsymbol{e}}(k+j)}{2}\end{aligned} \tag{52}$$

Next, take the time difference or incremental form of the Lyapunov function $\boldsymbol{L}_C(k)$ to yield

$$\begin{aligned}&\Delta\boldsymbol{L}_C(k+j) \\ &= \boldsymbol{L}_C(k+j+1) - \boldsymbol{L}_C(k+j) \\ &= \frac{1}{2}\sum_{j=d}^{N_P}\left(\hat{\boldsymbol{e}}^T(k+j+1)\hat{\boldsymbol{e}}(k+j+1) - \hat{\boldsymbol{e}}^T(k+j)\hat{\boldsymbol{e}}(k+j)\right) \\ &= \frac{1}{2}\sum_{j=d}^{N_P}\left(\left(\Delta\hat{\boldsymbol{e}}(k+j)\right)^T\left(2\hat{\boldsymbol{e}}(k+j) + \Delta\hat{\boldsymbol{e}}(k+j)\right)\right)\end{aligned} \tag{53}$$

Since $\hat{\boldsymbol{e}}(k+1) = \Delta\hat{\boldsymbol{e}}(k) + \hat{\boldsymbol{e}}(k)$ or $\Delta\hat{\boldsymbol{e}}(k) = \hat{\boldsymbol{e}}(k+1) - \hat{\boldsymbol{e}}(k)$, the future tracking error $\Delta\hat{e}(k+j)$ is found by

$$\begin{aligned}\Delta\hat{\boldsymbol{e}}(k+j) &\approx \Delta\boldsymbol{P}_C(k+j)\left[\frac{\partial\hat{\boldsymbol{e}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T \\ &= -\Delta\boldsymbol{P}_C(k+j)\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T \end{aligned} \tag{54}$$

where the matrix elements that are required to be convergent and not diverge to infinity are computed by the following

$$\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_c(k+j)} = \left(\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{k}_p(k+j)}, \frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{k}_i(k+j)}, \frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{k}_d(k+j)}\right)^T \tag{55}$$

By considering the predictive cost function (41), the increment of the updating parameter vector $\Delta\boldsymbol{P}_C(k+j)$ is obtained from the following formula

$$\begin{aligned}&\Delta\boldsymbol{P}_C(k+j) \\ &= -\eta_C(k)\frac{\partial E(k+j)}{\partial\boldsymbol{P}_C(k+j)} \\ &= -\eta_C(k)\frac{\partial E(k+j)}{\partial\hat{\boldsymbol{y}}(k+j)}\cdot\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)} \\ &= \eta_C(k)\left(\boldsymbol{r}(k+j) - \hat{\boldsymbol{y}}(k+j)\right)\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)} \end{aligned} \tag{56}$$

From (56), one obtains

$$\begin{aligned}&\Delta\hat{\boldsymbol{e}}(k+j) \\ &= \Delta\boldsymbol{P}_C(k+j)\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T \\ &= -\eta_C(k)\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T\hat{\boldsymbol{e}}(k+j) \end{aligned} \tag{57}$$

$$\begin{aligned}&\Delta\hat{\boldsymbol{e}}^T(k+j) \\ &= \left(\Delta\boldsymbol{P}_C(k+j)\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T\right)^T \\ &= -\eta_C(k)\hat{\boldsymbol{e}}^T(k+j)\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T \end{aligned} \tag{58}$$

Finally, $\Delta\boldsymbol{L}_C(k+j)$ is then obtained by substituting (57) and (58) into (53) and then one rewrites it to obtain

$$\begin{aligned}&\Delta L_C(k+j) \\ &= \frac{1}{2}\Delta\hat{\boldsymbol{e}}^T(k+j)\left(2\hat{\boldsymbol{e}}(k+j) + \Delta\hat{\boldsymbol{e}}(k+j)\right) \\ &\approx \frac{1}{2}\left(\Delta\boldsymbol{P}_C(k+j)\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T\right)^T \\ &\quad\left(2\hat{\boldsymbol{e}}(k+j) + \Delta\boldsymbol{P}_C(k+j)\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T\right) \\ &= -\frac{1}{2}\eta_C(k)\hat{\boldsymbol{e}}^T(k+j)\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T \\ &\quad\left(2 - \eta_C(k)\hat{\boldsymbol{e}}(k+j)\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T\right) \\ &= -\frac{1}{2}\eta_C(k)\hat{\boldsymbol{e}}^T(k+j)\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T \\ &\quad\left(2I - \eta_C(k)\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T\right)\hat{\boldsymbol{e}}(k+j) \end{aligned} \tag{59}$$

According to the Lyapunov stability theorem, similar to the previous arguments in Theorem 1, by denoting $\boldsymbol{B} = \frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T$, the asymptotically stability of the proposed controller is guaranteed if $2 - \eta\lambda_i > 0$ holds for any eigenvalues. Hence, we obtain:

$$\begin{aligned}&0 < \eta < 2/\lambda_{\max}(\boldsymbol{B}) \Rightarrow \\ &0 < \eta_C < 2\Big/\lambda_{\max}\left(\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T\right) \end{aligned} \tag{60}$$

which proves that the system is uniformly asymptotically stable. **Theorem 2** summarizes the main result.

Theorem 2: The MIMO ORFBLS-APPID controller is uniformly asymptotically stable in line with the controller learning rate η_c meets with the following sufficient conditions:

$$0 < \eta_c < \frac{2}{\lambda_{\max}\left(\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\left[\frac{\partial\hat{\boldsymbol{y}}(k+j)}{\partial\boldsymbol{P}_C(k+j)}\right]^T\right)} \tag{61}$$

V. CLOSED-LOOP STABILITY ANALYSIS OF OVERALL CONTROL SYSTEMS

In designing of the offered adaptive control system, the previous MIMO ORFBLS identifier and ORFBLS-APPID controller must be integrated together and then closely incorporated with an adaptive control algorithm. In doing so, this section will advance a real-time ORFBLS-based approach for system identification and adaptation. Specifically, since the theory of the MIMO ORFBLS identifier has been presented in detail along with its learning algorithm and convergence analysis, so the identifier asymptotically convergence will be maintained over time, and no single parameters (25) diverge. After realizing this system identifier, the MIMO adaptive predictive control algorithm is then executed by applying Jacobian transformations as a bridge between the MIMO ORFBLS identifier and adaptive predictive PID controller, where the gains matrices of PID are updated online over time [22], [27], [39]. The detailed procedures are elaborated in the following eight steps:

Step 1: Analyze and retain the vector of $\boldsymbol{u}(k)$ and $\boldsymbol{y}(k)$.

Step 2: Find d, n_u and n_y drawn from the system model.

Step 3: Define K and η to be positive and real numbers, and initialize $\boldsymbol{k}_p$, $\boldsymbol{k}_i$ and $\boldsymbol{k}_d$, and parameter vector $\boldsymbol{P} = \left[\boldsymbol{W}_f\boldsymbol{W}_h\,\boldsymbol{W}_e\boldsymbol{\beta}_e\boldsymbol{A}_e\boldsymbol{A}_f\right]^T$ using small random numbers.

Step 4: Estimate $\hat{y}(k)$ from the ORFBLS identifier (14).

Step 5: Compute $\boldsymbol{u}(k) = \Delta\boldsymbol{u}(k) + \boldsymbol{u}(k-1)$ using (40).

Step 6: Adjust the ORFBLS identifier by following (25) and (26) to confirm the ORFBLS identifier learning rate η.

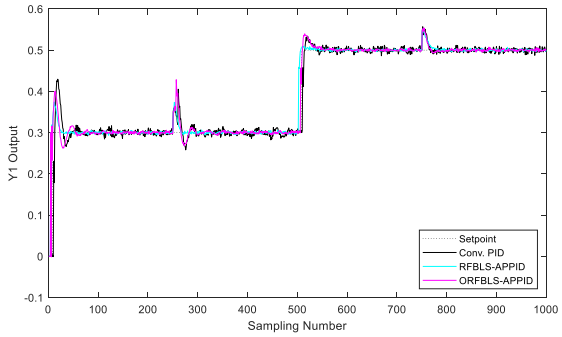

FIGURE 4. Comparative setpoint tracking for y_1 in Example 1.

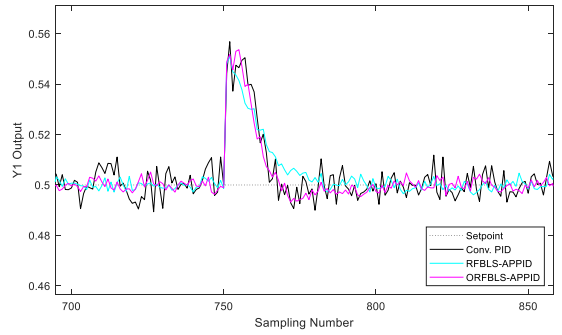

FIGURE 5. Comparative setpoint tracking for y_1 in Example 1 (explosive chart).

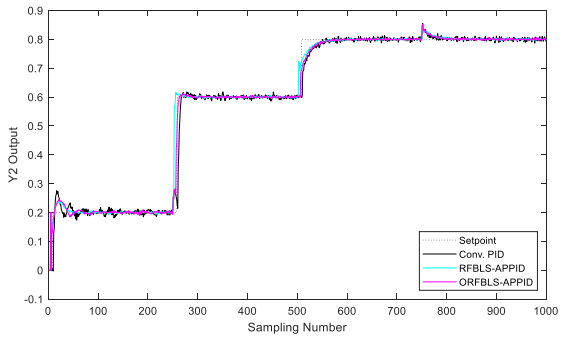

FIGURE 6. Comparative setpoint tracking results for y_2 in Example 1.

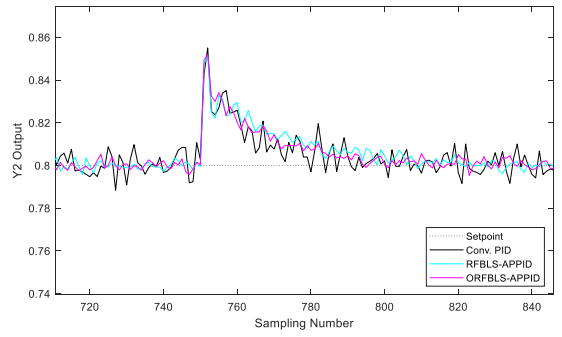

FIGURE 7. Comparative setpoint tracking for y_2 in Example 1 (explosive chart).

Step 7: Tune the gain matrices of the ORFBLS-APPID controller by following (49) - (51) and (61) to confirm the controller learning rate η_c.

Step 8: Repeat steps 4 through 7.

Thus, the main result of this algorithm is delineated in Theorem 3.

Theorem 3: Assume that the MIMO ORFBLS identification satisfies **Theorem 1** and the adaptive predictive MIMO ORFBLS-PID controller obeys **Theorem 2**. Then the overall closed-loop system implemented by the real-time identification and control algorithm is uniformly asymptotically stable.

Proof of Theorem 3:

To show the uniformly asymptotical stability of the overall closed-loop system incorporated in the real-time system identification and control algorithm, one defines the d-step-ahead tracking error $\boldsymbol{e}(k+d) = \boldsymbol{r}(k+d) - \boldsymbol{y}(k+d)$, which is rewritten by $\boldsymbol{e}(k+j) = \big(\boldsymbol{r}(k+j) - \hat{\boldsymbol{y}}(k+j)\big) - \big(\hat{\boldsymbol{y}}(k+j) - \boldsymbol{y}(k+j)\big)$. Obviously, the subsequent inequalities always hold, i.e., $\|\boldsymbol{e}(k+d)\|_2 \leq \big\|\boldsymbol{r}(k+j) - \hat{\boldsymbol{y}}(k+j)\big\|_2 + \big\|\hat{\boldsymbol{y}}(k+j) - \boldsymbol{y}(k+j)\big\|_2$ and $\|\boldsymbol{e}(k+d)\|_2^2 \leq 2\Big(\big\|\boldsymbol{r}(k+j) - \hat{\boldsymbol{y}}(k+j)\big\|_2^2 + \big\|\hat{\boldsymbol{y}}(k+j) - \boldsymbol{y}(k+j)\big\|_2^2\Big)$. Following the proof procedures previously outlined in Sections III and IV, one chooses that $L(k) = 2\left(L_M(k) + L_C(k)\right)$ and then obtains that $\Delta L(k) = 2\left(\Delta L_M(k) + \Delta L_C(k)\right)$. Therefore, $\Delta L\,(k)$ is shown to be negative definite if the uniformly asymptotically convergent condition (26) of the ORFBLS identifier and uniformly asymptotically stable condition (61) of the ORFBLS-APPID controller are held simultaneously.

Remark 1: In adopting **Theorem 3**, there are five rules for choosing the main parameters before conducting any numerical simulations or experiments. First, the system inputs with order n_u and system output order n_y are configured to match or exceed the controller output u and plant output y. Second, the predictive output horizon N_P should be configured to match or exceed the delay time d. In the presence of high uncertainties, the time delay d configured to be not more than N_P. Third, the rule expansion coefficient K value is defined as a small number to lessen the load of the numerical computation. Fourth, in order to satisfy the inequality condition (26), the identifier learning rate is reduced in the time of deploying numerical analysis and experimental study. Fifth, the control learning rate η_c obligated to comply with the inequality (61). In real applications, η_c is configured as small as possible.

VI. COMPARATIVE SIMULATIONS AND DISCUSSION

A. PERFORMANCE EVALUATION INDEXES

To accurately assess the effectiveness of the proposed MIMO adaptive control method, this section applies the performance evaluation indexes similar to [3], [16], [26], and [40], and provides their detailed formulas as follows;

$$Max\ Error = \max \|\boldsymbol{r}(k) - \boldsymbol{y}(k)\|_2 \tag{62}$$

$$RMSE = \sqrt{\sum_{k=1}^{S} \|\boldsymbol{r}(k) - \boldsymbol{y}(k)\|_2^2 \Big/ S} \tag{63}$$

$$ISE = \sum_{k=1}^{S} \|\boldsymbol{r}(k) - \boldsymbol{y}(k)\|_2^2 \tag{64}$$

$$IAE = \int_0^{1000T} \|\boldsymbol{e}(t)\| dt \cong T \sum_{k=1}^{S} \|\boldsymbol{r}(k) - \boldsymbol{y}(k)\|_2 \quad (65)$$

$$ITAE = \sum_{k=1}^{S} \|\boldsymbol{r}(k) - \boldsymbol{y}(k)\|_2 T \cdot k \quad (66)$$

where the vector $\boldsymbol{r}(k)$ represents the designated setpoint, and $\boldsymbol{y}(k)$ stands for the real system output at the sampling instances k. The variable T signifies the sampling period and S means the total number of samples taken.

B. EXAMPLE 1

The first evaluative example deploys the two-by-two MIMO system model characterized by time-varying dynamics, uncertainty, and exhibiting strong nonlinearity and coupling, akin to the model described in [41] and [42]. The complex model is written by the following;

$$\begin{aligned} y_1(k) &= \begin{cases} \dfrac{a_1(k) y_1(k-1) + u_1(k-1-d)}{1 + y_1^2(k-1)} \\ + \dfrac{0.5 * u_2(k-1-d)}{1 + u_2^2(k-1-d)} \end{cases} + \varepsilon(k) + v(k) \\ y_2(k) &= \begin{cases} \dfrac{a_2(k) y_2(k-1) + u_2(k-1-d)}{1 + y_2^2(k-1)} \\ + \dfrac{0.2 * u_1(k-1-d)}{1 + u_1^2(k-1-d)} \end{cases} + \varepsilon(k) + v(k) \end{aligned} \quad (67)$$

where the slowly time-varying factors $a_1(k) = 1.2\left(1 - 0.8e^{-0.5k}\right)$ and $a_2(k) = 1.2\left(1 - 0.8e^{-0.1k}\right)$ evolve the dynamics of the system over time. The system time delay d is configured to 0. Additionally, $\varepsilon(k)$ represents the presence of Gaussian white noise, while $v(k)$ characterizes the load disturbances that influence the system behavior. In this example, $v(k)$ is set as follows:

$$v(k) = \begin{cases} 0, & 0 < k \le 250 \\ 0.05, & 250 < k \le 750 \\ 0.1, & 750 < k \le 1000 \end{cases} \quad (68)$$

Afterward, the setpoint vector $\boldsymbol{r}(k)$, including two distinct setpoints $r_1(k)$ and $r_2(k)$ designed uniquely for the outputs of each system are described as follows;

$$\begin{aligned} r_1(k) &= \begin{cases} 0, & 0 < k \le d \\ 0.3, & d < k \le 500 \\ 0.5, & 500 < k \le T \end{cases}, \\ r_2(k) &= \begin{cases} 0, & 0 < k \le d \\ 0.2, & d < k \le 250 \\ 0.6, & 250 < k \le 500 \\ 0.8, & 500 < k \le T \end{cases} \end{aligned} \quad (69)$$

Employing the ORFBLS algorithm for system identification, the simulation utilizes the identifier adaptation learning rate $\eta = 0.7$, the maximum predictive horizon $N_p = 7$ and the controller adaptation learning rate $\eta_c = 1$. Additionally,

TABLE 1. Comparative performance indexes of different controllers in Example 1.

Controller	Max Error	RMSE	ISE	IAE	ITAE
Conventional PID	0.46848	0.04419	1.38464	15.49712	5224.92
RFBLS-APPID	0.46242	0.04012	1.13781	12.73958	4623.36
Proposed ORFBLS-APPID	0.45550	0.03304	0.79190	10.05719	3615.57

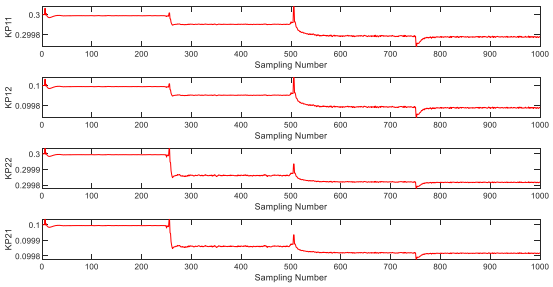

FIGURE 8. Time evolutions of k_P gains in Example 1.

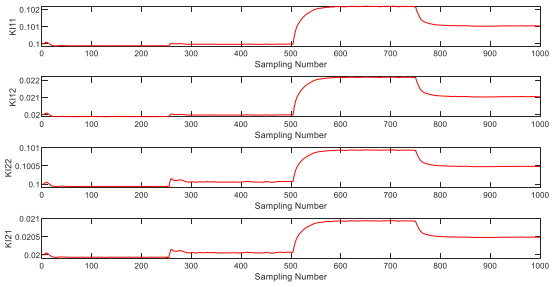

FIGURE 9. Time evolutions of k_I gains in Example 1.

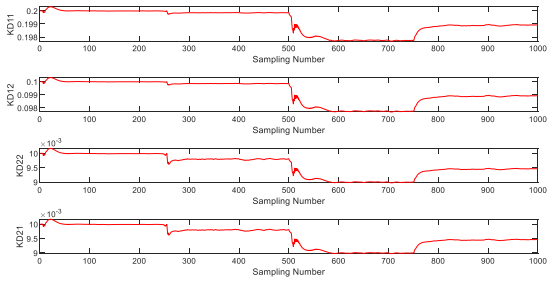

FIGURE 10. Time evolutions of k_d gains in Example 1.

the key architecture specifications of the ORFBLS identifier are designed to work with criteria in the following; we set the fuzzy rules governing identifier $K = 4$, fuzzy membership function $m = 4$, the system identifier input $M_{ORFBLS-APPID} = 5$, enhancement node number $L = 4$ to boost the generalization ability, and lastly the total sampling number $T = 1000$. Then, the settings of the two-by-two PID gain matrices tailored explicitly for both independent PID control loops are initialized as follows;

$$k_p = \begin{bmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{bmatrix}, k_i = \begin{bmatrix} 0.1 & 0.01 \\ 0.1 & 0.1 \end{bmatrix}, k_d = \begin{bmatrix} 0.2 & 0.1 \\ 0.01 & 0.01 \end{bmatrix} \quad (70)$$

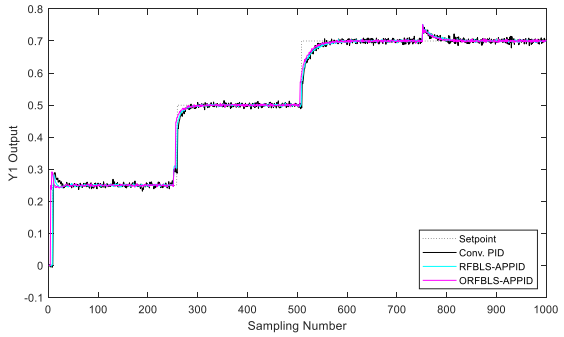

FIGURE 11. Comparative setpoint tracking for y_1 in Example 2.

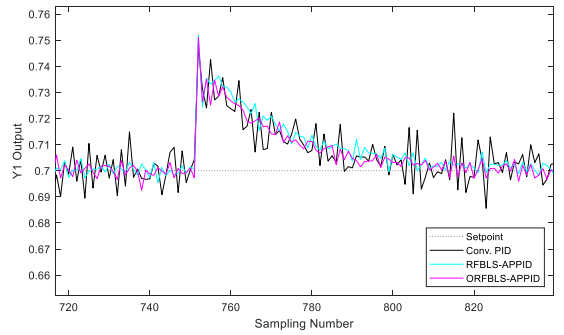

FIGURE 12. Comparative setpoint tracking for y_1 in Example 2 (explosive chart).

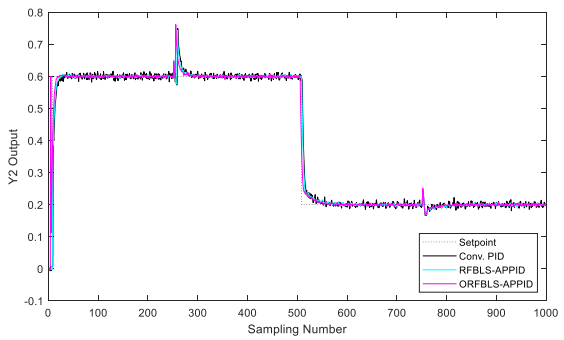

FIGURE 13. Comparative setpoint tracking for y_2 in Example 2.

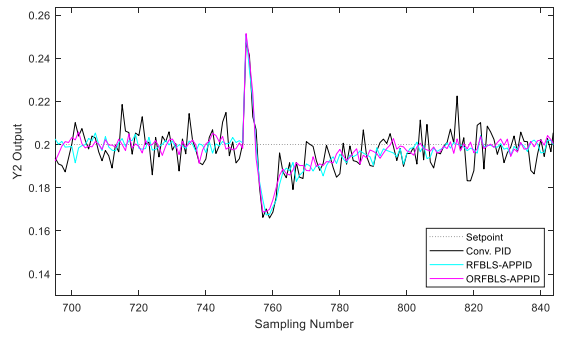

FIGURE 14. Comparative the setpoint tracking for y_2 in Example 2 (explosive chart).

The setpoint tracking capabilities and disturbance rejection performance are analyzed across controllers by inspecting the output responses in **Fig. 4** and **Fig. 6**, respectively. Evidently, the MIMO ORFBLS-APPID controller achieves superior control performance and robustness properties compared to the conventional fixed-gain MIMO PID and MIMO RFBLS-APPID methods, especially in response to setpoint changes.

Further investigation of the explosive setpoint tracking results in **Fig. 5** and **Fig. 7** quantitatively verifies the enhanced adaptation of the proposed controller architecture where the load disturbance rejection ability slowly converges the error to zero with less overshoots and less oscillations. Moreover, as shown in **Figs. 8 – 10**, the time evolutions of the PID gains demonstrate how the MIMO ORFBLS-APPID approach automatically tunes parameters over time in response to internal model updates and external errors.

Finally, after carrying out numerical comparative simulations of Example 1, the tabulated performance metrics in **Table 1** numerically confirm the benefits of the MIMO ORFBLS-APPID controller for enhancing multi-variable processes through online learning and adaptive control. Compared to traditional PID and RFBLS-APPID techniques, the offered controller significantly showcases superior transient performance, rise time, peak errors, and disturbance rejection capabilities. This first example thus validates the proposed control method as an effective methodology.

C. EXAMPLE 2

Expanding on the preliminary simulations, another challenging test system adapted from [41] and [42] is taken as the second multivariate simulation example, in order to exemplify augmented complexity through a customized nonlinear time-varying dynamics system model exhibiting exacerbated coupling effects. This modified system model is described by

$$\begin{aligned} y_1(k+1) &= \frac{y_1(k)}{1+y_1^2(k)} + u_1(k-d) + \varepsilon(k) + v(k) \\ y_2(k+1) &= \frac{y_1(k)y_2(k)}{1+y_2^2(k)} + u_2(k-d) + \varepsilon(k) + v(k) \end{aligned} \quad (71)$$

with d is configured to 1 and the uphold load disturbance $v(k)$ is set as in (68) for Example 1.

Mirroring the dual setpoint trajectory structure imposed in Example 1, two independent reference inputs $r_1(k)$ and $r_2(k)$ dictate the desired responses over the simulation run-time for the multivariate process, and they are mathematically constructed as

$$\begin{aligned} r_1(k) &= \begin{cases} 0, & 0 < k \le d \\ 0.25, & d < k \le 250 \\ 0.5, & 250 < k \le 500 \\ 0.7, & 500 < k \le 1000, \end{cases} \\ r_2(k) &= \begin{cases} 0, & 0 < k \le d \\ 0.6, & d < k \le 500 \\ 0.2, & 500 < k \le 1000 \end{cases} \end{aligned} \quad (72)$$

Eventually, mirroring the parameters stated in precedent Example 1, comparative simulations in Example 2 continue to

leverage the ORFBLS for adaptive control. However, the key identifier and controller learning rates η and η_c are attenuated to 0.6 and 0.1 respectively. Moreover, the initial two-by-two PID gain matrices are set in the following

$$k_p = \begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.1 \end{bmatrix}, k_i = \begin{bmatrix} 0.07 & 0.01 \\ 0.2 & 0.3 \end{bmatrix}, k_d = \begin{bmatrix} 0.02 & 0.01 \\ 0.01 & 0.01 \end{bmatrix} \tag{73}$$

Eventually, examining the output response plots in **Fig. 11** and **Fig. 13** verifies the superior setpoint tracking and attenuated oscillations from the MIMO ORFBLS-APPID architecture in comparison with the baseline MIMO PID and RFBLS-APPID controllers. Moreover, by looking into the explosive setpoint tracking charts in **Fig. 12** and **Fig. 14**, the proposed controller further exhibits faster rise times and reduced overshoot with better noise cancellation. Notably, this enhanced performance emerges despite more aggressive setpoint changes and exacerbated nonlinear plant effects in Example 2. Overall, the simulation results reveal the true adaptivity of the MIMO ORFBLS-APPID controller in which these PID gains in **Figs. 15 - 17** are continually tuned online to adapt to the complex dynamics of the system model.

Finally, after carrying out comparative simulations for Example 2, the comparative performance indexes presented in Table 2 statistically reinforce the advantages of the proposed control scheme. Comparing MIMO ORFBLS-APPID against traditional methodologies numerically proves significant improvements in responsiveness, steady-state accuracy, and disturbance rejection capabilities. Therefore, the comparative simulation results in Examples 1 and 2 collectively validate the presented adaptive ORFBLS learning configuration as an impactful advancement for multivariable processes through the exploitation of output recurrent knowledge.

D. EXAMPLE 3: MIMO TEMPERATURE CONTROL OF AN EXTRUSION BARREL IN A PLASTIC INJECTION MOLDING MACHINE

The third example is aimed to investigate the effectiveness and merits of the proposed control approach by using a model of an experimental setup which is an industrial extrusion barrel in an injection molding machine. Such a system model is established by the following:

$$y(k+1) = Ay(k) + B_0 u(k-d) + B_1 u(k-d-1) + \varepsilon(k) \tag{74}$$

where the time delay d is 6 and following [24] and [43], two coupling effect matrices, $diag\{0.001, 0.001, 0.001, 0.001\}$ and $diag\{0.0005, 0.0005, 0.0005, 0.0005\}$ are attained to both matrices B_0 and B_1, respectively. Also, considering insignificant inter-zone thermal coupling observed across zones, the matrix A can justifiably be reduced to a diagonal matrix $diag\{1, 1, 1, 1\}$.

Similar to Examples 1 and 2, comparative simulations for Example 3 employ the previously aligned ORFBLS parameters, but with a few exceptions. These exceptions include

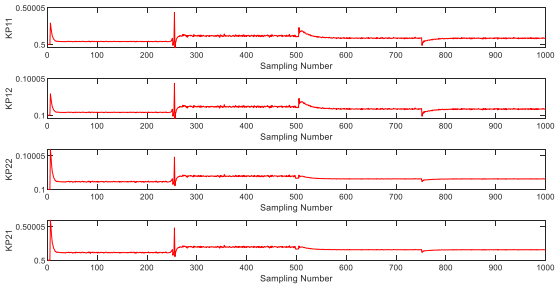

FIGURE 15. Time evolutions of k_p gains in Example 2.

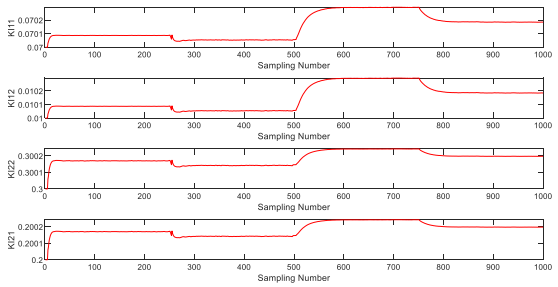

FIGURE 16. Time evolutions of k_i gains in Example 2.

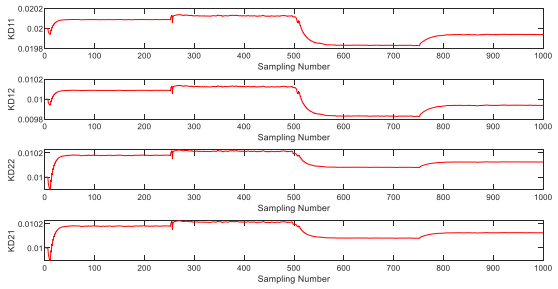

FIGURE 17. Time evolutions of k_d gains in Example 2.

that the identifier learning rate is renewed as $\eta = 0.5$, the controller learning rate $\eta_C = 0.1$, and the total sampling number $T = 2000$. Next, by following the zone configuration in **Fig. 30**, the two-level setpoint vector for all the zones in the heating barrel is set as follows;

$$\boldsymbol{r} = \begin{cases} [130°\text{C}, 120°\text{C}, 110°\text{C}, 100°\text{C}] & 0 < k \leq 1000 \\ [200°\text{C}, 190°\text{C}, 180°\text{C}, 170°\text{C}] & 1000 < k \leq 2000 \end{cases} \tag{75}$$

Assessing the controller performance to follow the desired system response (75), the initial PID gains parameters are configured in the sequel:

$$\begin{aligned} k_p &= diag\,(32.10\ 25.11\ 20.50\ 16.71) \\ k_i &= diag\,(0.010\ 0.010\ 0.010\ 0.010) \\ k_d &= diag\,(180.0\ 200.0\ 200.0\ 200.0) \end{aligned} \tag{76}$$

Analyzing the extensive output trajectories for each control zone in **Figs. 18 - 25** verifies precise setpoint tracking done

TABLE 2. Comparative performance indexes of different controllers in Example 2.

Controller	Max Error	RMSE	ISE	IAE	ITAE
Conventional PID	0.65707	0.04017	1.19514	12.93409	5691.91
RFBLS-APPID	0.65009	0.03895	1.14123	11.08852	4297.43
Proposed ORFBLS-APPID	0.54917	0.03554	1.06338	9.20361	3873.08

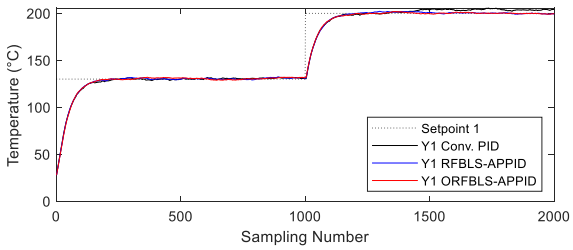

FIGURE 18. Comparative setpoint tracking for y_1 in Example 3.

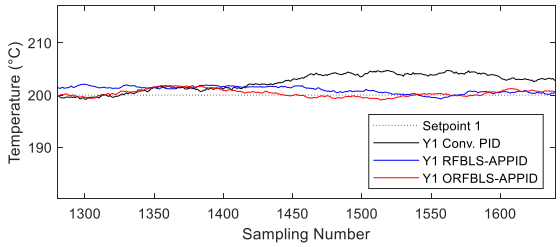

FIGURE 19. Comparative setpoint tracking for y_1 in Example 3 (explosive chart).

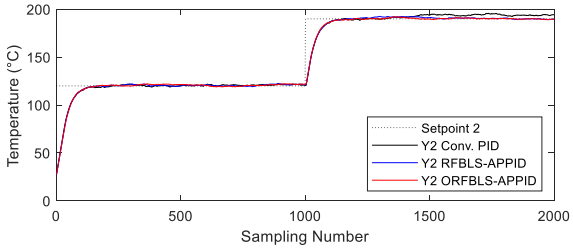

FIGURE 20. Comparative setpoint tracking for y_2 in Example 3.

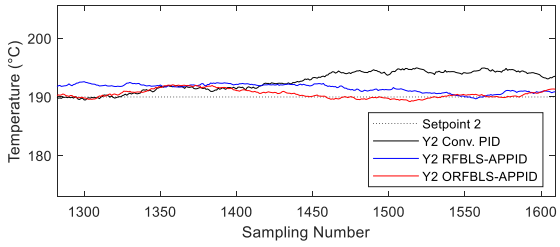

FIGURE 21. Comparative setpoint tracking for y_2 in Example 3 (explosive chart).

by the MIMO ORFBLS-APPID architecture amid the interacting multivariable temperature dynamics. As can be seen in **Figs. 18 - 25,** minimal overshoot and rapid stabilization

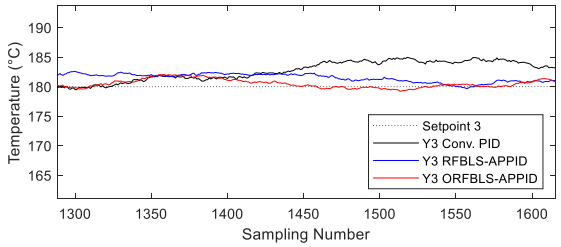

FIGURE 22. Comparative setpoint tracking for y_3 in Example 3.

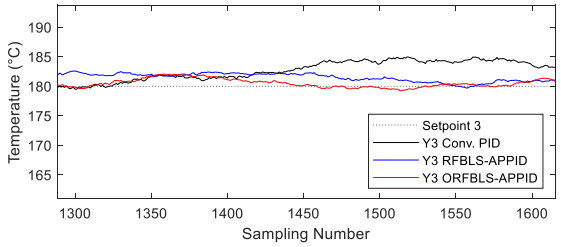

FIGURE 23. Comparative setpoint tracking for y_3 in Example 3 (explosive chart).

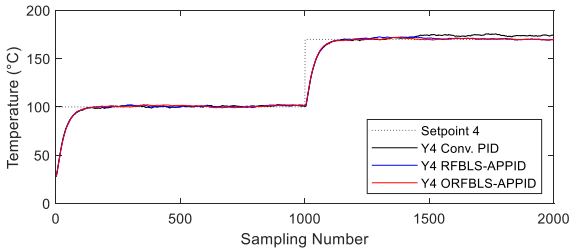

FIGURE 24. Comparative setpoint tracking for y_4 in Example 3.

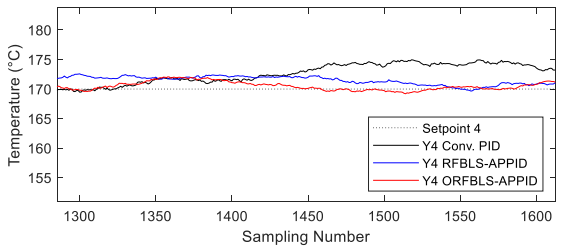

FIGURE 25. Comparative setpoint tracking for y_4 in Example 3 (explosive chart).

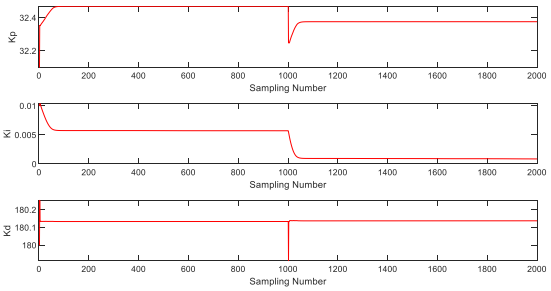

FIGURE 26. Time evolutions of k_p, k_i and k_d of zone 1 in Example 3.

further showcase the adaptive capabilities of the developed controller, even under long and nonlinear transients.

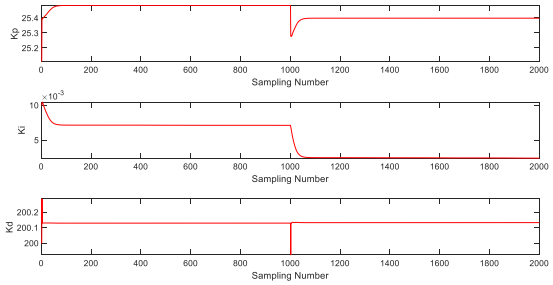

FIGURE 27. Time evolutions of k_p, k_i and k_d of zone 2 in Example 3.

Interestingly, while the PID gains evolve continuously in **Figs. 26 – 29,** to account for complex heating interactions, the proposed method automatically identifies the appropriate control direction by successfully tuning the parameters of each zone without the fragility of manual tuning. The self-tuning results from this approach elucidate the resilience to sensor noise, the superior noise cancellation capabilities to load disturbances, and the adaptability to parameter changes evidenced across these highly accurate response plots.

In conclusion, the tabulated metrics presented in **Table 3** serve as quantitative measures to illustrate the effectiveness of the proposed MIMO ORFBLS-APPID controller in regulating the controller to achieve better system responses. These remarkable results numerically confirm the advantages of the MIMO ORFBLS-APPID framework across a range of multivariable coupling and nonlinearity scenarios. By embedding adaptive ORFBLS logics and recursive learning capabilities directly into the feedback structure, the adaptivity property enables the MIMO ORFBLS-APPID controller to outperform conventional MIMO fixed gains PID controller and RFBLS-based adaptive PID controller. The proposed control scheme thus pushes performance improvements in setpoint tracking, disturbance rejection, rise times, overshoot, and long-term accuracy. Statistically and dynamically, these simulations comprehensively validate the strengths of proposed adaptive learning control systems in balancing regulation across complex interdependent processes.

VII. APPLICATION TO MIMO TEMPERATURE CONTROL OF THE EXTRUSION BARREL: EXPERIMENTAL RESULTS AND DISCUSSION

In this section, an experimental validation on a laboratory injection molding extruder is conducted to provide key real-world qualifications of the suggested multivariable controller prior to industrial deployment. The customized extrusion barrel features four decoupled heating zones, replicating target polymeric melt systems where multiple inputs, sensors, and complex thermal behaviors necessitate advanced control techniques like the integration of the proposed ORFBLS-APPID approach.

As depicted in **Fig. 30**, the heating elements f_1 to f_4 correspond to zones 1 to 4 respectively, enabling either

TABLE 3. Comparative performance indexes of different controllers in Example 3.

Controller	Max Error	RMSE	ISE	IAE	ITAE
Conventional PID	161.642	17.48413	299274	10058.85	6701110
RFBLS-APPID	157.543	17.12475	290435	11000.25	6596070
Proposed ORFBLS-APPID	155.092	17.00733	289703	9020.41	6048150

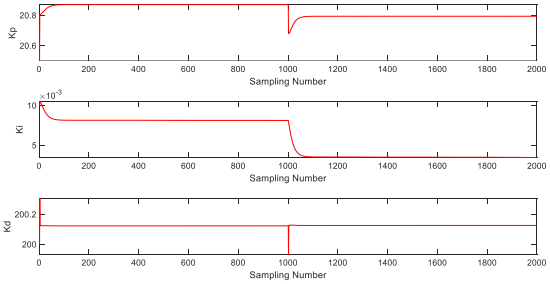

FIGURE 28. Time evolutions of k_p, k_i and k_d of zone 3 in Example 3.

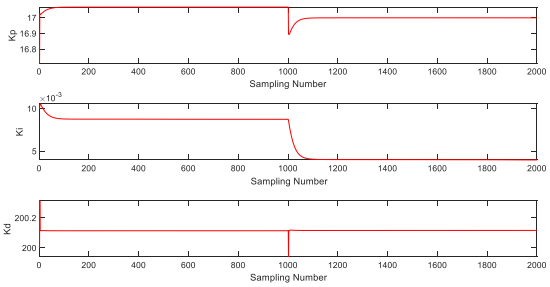

FIGURE 29. Time evolutions of k_p, k_i and k_d of zone 4 in Example 3.

independent SISO or coupled MIMO control architectures. On the other hand, by following **Fig. 31,** outside the main controller, three independent blocks with different functions and system objectives are assembled and systemized. To convert the three-phase 220 VAC energy from the power source, four solid state relays (SSRs) are used with the control signal received from the main controller unit in the form of a duty cycle in a Pulse Width Modulation (PWM) signal at 1 Hz of frequency. SPI communication allows the K-type thermocouples with MAX6675 signal processors to transmit zone temperature readings for feedback comparisons against shifting setpoints [44], [45].

The actual view of the laboratory-built experimental setup is depicted in **Fig. 32.** The extrusion barrel (pile cylinder-like round metal) was constructed with four heating elements in two forms of size, three larger heating elements f_2, f_3, and f_4 were equally installed in the barrel and one smaller heating element f_1 was installed at the tip of the extrusion barrel. In this real application, the polymer material enters from the left side of the figure and is forced out to the right side using one or two screws. The four SSRs (FOTEK SSR-25 DA) were

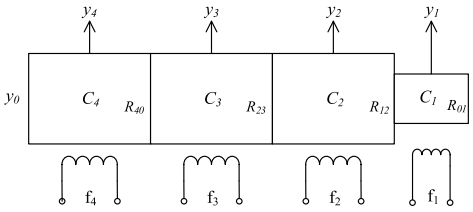

FIGURE 30. Physical model of extrusion barrel.

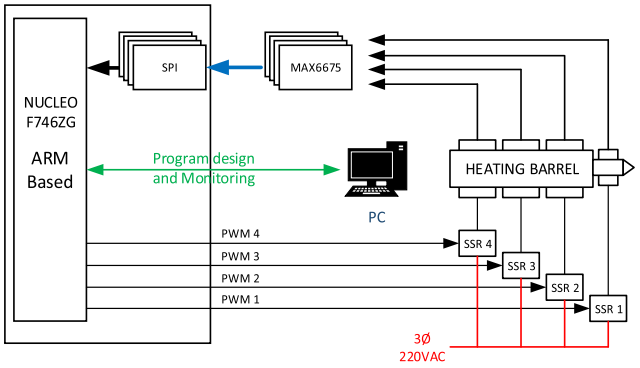

FIGURE 31. Schematic diagram of temperature control for the extrusion barrel in an injection molding machine.

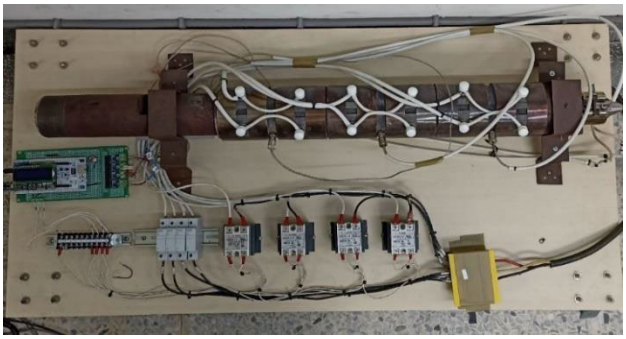

FIGURE 32. Actual view of the laboratory-built experimental setup.

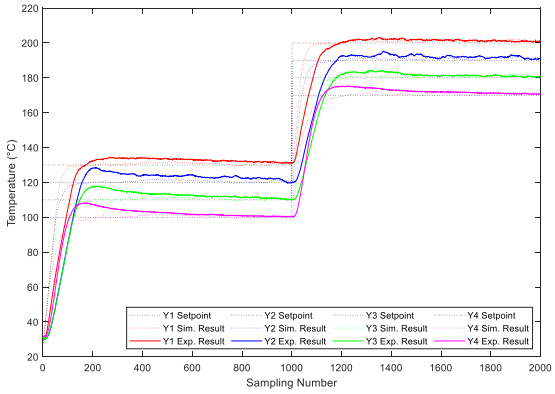

FIGURE 33. Comparative setpoint tracking results of Example 3 and experimental result.

placed just below the barrel extruder and the main controller (STM32-F746ZG) was placed on the left side between the barrel and SSRs [46], [47].

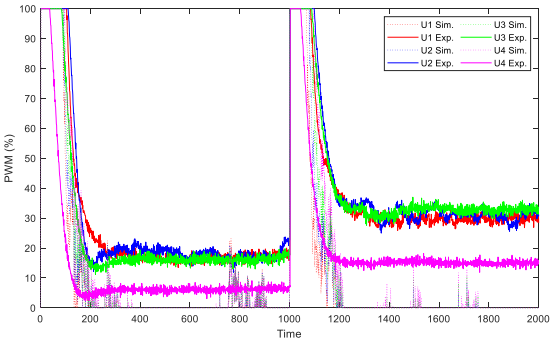

FIGURE 34. Comparative controller output of Example 3 and experimental result.

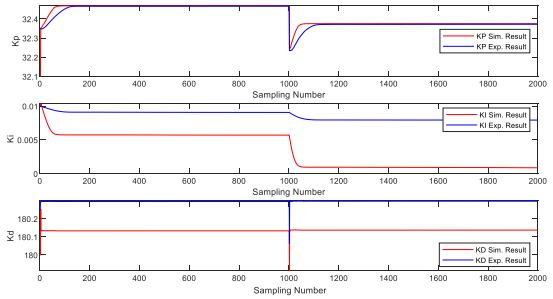

FIGURE 35. Comparison of the PID gains evolutions of Example 3 and experimental results in Zone 1.

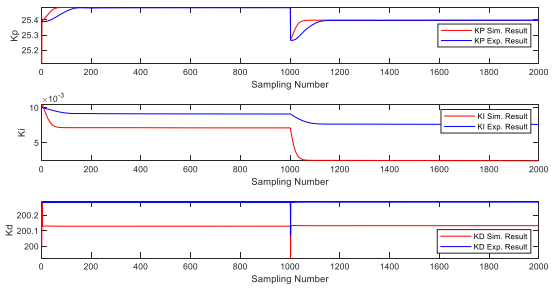

FIGURE 36. Comparison of the PID gains evolutions of Example 3 and experimental results in Zone 2.

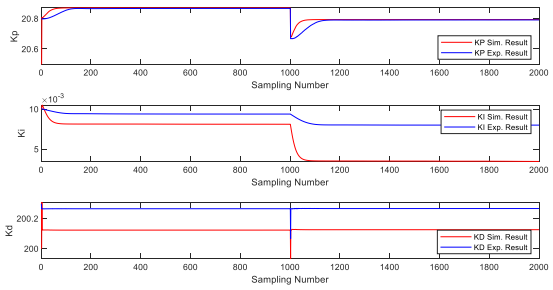

FIGURE 37. Comparison of the PID gains evolutions of Example 3 and experimental results in Zone 3.

Aligning the comparative simulations for Example 3, this experiment utilized identical control parameters and settings to allow direct benchmarking between simulation and physical systems. Later figures visually show the comparisons, highlighting some deviations between the simulation and

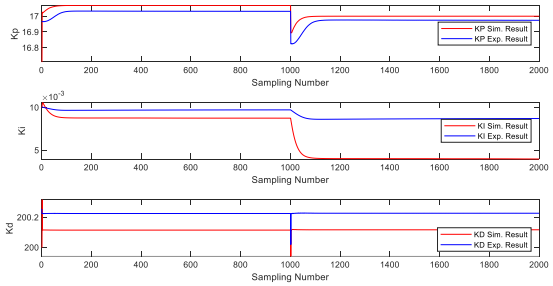

FIGURE 38. Comparison of the PID gains evolutions of Example 3 and experimental results in Zone 4.

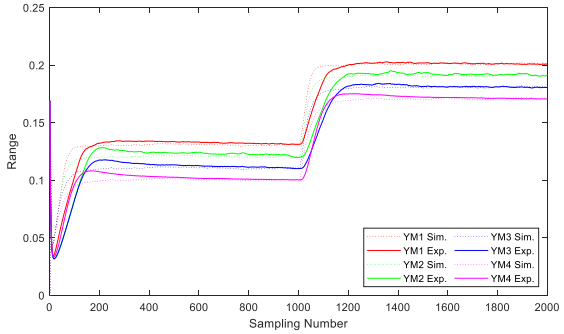

FIGURE 39. Comparison of the ORFBLS predictive outputs of Example 3 and experimental results.

experimentation both in setpoint tracking per **Fig. 33** and control output in **Fig. 34**. Discrepancies likely originate from the simplified digital model struggling to fully replicate extruder dynamics like couplings. Specifically, up-down deviations in zones 2 and 3 indicate unmodeled interactions induce some physical response lags absent in simulations. Still, strong overall correlation and tracking validate the practical applicability, with minor gaps informing refinements to tighten predictions moving forward.

Figs. 35 - 38 compare PID gain evolutions between comparative simulations in Example 3 and experimental results across all four zones. Visually, the data series followed similar trajectories, evolving responsively to follow the changing of the desired system responses. This adaptive correlation despite real-world variations reinforces accuracy in the modeling and identification powering self-tuning activities. Additionally, **Fig. 39** compares the ORFBLS predictive outputs between the simulation result obtained in Example 3 and the result obtained by the experimental study, showing almost consistency of the simulation and experimental results.

Finally, realizing the result from the experimental study depicted in **Fig. 33** to **Fig. 39**, we can conclude that in such a windy environment, the outcomes reveal that the system maximum overshoots are less than 3°C while the steady-state errors remained within 1°C. Aside from that, even though the comparison results show some differences, it can still be concluded that this experimental study result has successfully proven that the developed MIMO ORFBLS-APPID controller is applicable and practicable, and its controlled system response can be effectively adapted to the changing setpoints and dynamic environments.

VIII. CONCLUSION AND FUTURE WORK

For a class of MIMO systems, this paper has presented a novel control approach using an output recurrent fuzzy broad learning system (ORFBLS) for developing an adaptive predictive PID controller. The developed controller has been constructed by leveraging the MIMO ORFBLS identifier for online system and parameter identification, and the adaptive MIMO ORFBLS-APPID controller to dynamically update matrix gain parameters ($\boldsymbol{k}_p, \boldsymbol{k}_i$ and $\boldsymbol{k}_d$) to ensure precise setpoint tracking and disturbance rejection. Three theorems have been established to find the sufficient conditions of the learning rates both for the MIMO ORFBLS identifier and overall MIMO ORFBLS-APPID controller. Through three comparative simulations on two illustrative MIMO control processes and one physical model of an extrusion barrel in an injection molding machine, the developed ORFBLS-APPID method has been testified effective and superior by comparing to the conventional fixed-gain MIMO PID controller and MIMO RFBLS-APPID controller. One experimental validation on a laboratory-built extrusion barrel has affirmed the practicality and adaptability of the proposed controller to real-world scenarios.

In future work, the extended version of the proposed controller combined with a deep learning algorithm such as LSTM [44] and reinforcement learning would deserve further investigation, and its application to advanced semiconductor packaging equipment would show the applicability of such a new controller.

REFERENCES

[1] R. C. Panda, *Introduction to PID Controllers: Theory, Tuning and Application to Frontier Areas*. Rijeka, Croatia: InTech, 2012.
[2] P. Vega, C. Prada, and V. Aleixandre, "Self-tuning predictive PID controller," *IEE Proc. D Control Theory Appl.*, vol. 138, no. 3, p. 303, May 1991, doi: 10.1049/ip-d.1991.0041.
[3] K. Ogata, *Modern Control Engineering*, 5th ed. London, U.K.: Pearson, 2017.
[4] C.-C. Tsai and Y.-L. Chang, "Self-tuning PID control using recurrent wavelet neural networks," in *Proc. IEEE Int. Conf. Syst. Man, Cybern. (SMC)*, Oct. 2012, pp. 3111–3116, doi: 10.1109/ICSMC.2012.6378269.
[5] T. Yamamoto, S. Omatu, and M. Kaneda, "A design method of self-tuning PID controllers," in *Proc. Amer. Control Conf. (ACC)*, Jun. 1994, pp. 3263–3267, doi: 10.1109/acc.1994.735178.
[6] J. Lee, P. H. Chang, B. Yu, and M. Jin, "An adaptive PID control for robot manipulators under substantial payload variations," *IEEE Access*, vol. 8, pp. 162261–162270, 2020, doi: 10.1109/ACCESS.2020.3014348.
[7] O. Alshammari, M. N. Mahyuddin, and H. Jerbi, "An advanced PID based control technique with adaptive parameter scheduling for a nonlinear CSTR plant," *IEEE Access*, vol. 7, pp. 158085–158094, 2019, doi: 10.1109/ACCESS.2019.2948019.
[8] G. Dyanamina and S. K. Kakodia, "Adaptive neuro fuzzy inference system based decoupled control for neutral point clamped multi level inverter fed induction motor drive," *Chin. J. Electr. Eng.*, vol. 7, no. 2, pp. 70–82, Jun. 2021, doi: 10.23919/CJEE.2021.000017.
[9] M. Steinberger, M. Horn, and A. Ferrara, "Adaptive control of multivariable networked systems with uncertain time delays," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 489–496, Jan. 2022, doi: 10.1109/TAC.2021.3083563.

[10] Y. Zhan, S. Sui, and S. Tong, "Adaptive fuzzy decentralized dynamic surface control for fractional-order nonlinear large-scale systems," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 8, pp. 3373–3383, Aug. 2022, doi: 10.1109/TFUZZ.2021.3114746.

[11] S. Sui, Y. Zhan, J. Jin, C. L. P. Chen, and S. Tong, "Adaptive fuzzy decentralized control for fractional-order nonlinear large-scale systems with unmodeled dynamics," *IEEE Access*, vol. 9, pp. 142594–142604, 2021, doi: 10.1109/ACCESS.2021.3119611.

[12] Y. Zhan and S. Tong, "Adaptive fuzzy output-feedback decentralized control for fractional-order nonlinear large-scale systems," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12795–12804, Dec. 2022, doi: 10.1109/TCYB.2021.3088994.

[13] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985, doi: 10.1109/TSMC.1985.6313399.

[14] 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, doi: 10.1109/TCYB.2018.2857815.

[15] X. Gong, T. Zhang, C. L. P. Chen, and Z. Liu, "Research review for broad learning system: Algorithms, theory, and applications," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8922–8950, Sep. 2022, doi: 10.1109/TCYB.2021.3061094.

[16] 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.

[17] J.-F. Qiao, Y. Hou, L. Zhang, and H.-G. Han, "Adaptive fuzzy neural network control of wastewater treatment process with multiobjective operation," *Neurocomputing*, vol. 275, pp. 383–393, Jan. 2018, doi: 10.1016/j.neucom.2017.08.059.

[18] 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 dynamic systems," in *Proc. Int. Conf. Fuzzy Theory Its Appl. (iFUZZY)*, Nov. 2020, pp. 1–6, doi: 10.1109/iFUZZY50310.2020.9297812.

[19] Y. C. Li, "Intelligent auto-tuning of PID controllers using fuzzy broad learning system for tool-grinding servo control systems," M.S. thesis, Dept. Elect. Eng., Nat. Chung Hsing Univ., Taichung, Taiwan, 2019.

[20] 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.

[21] C. H. Yang, C. C. Tsai, and F. C. Tai, "Adaptive nonlinear PID control using RFBLS for digital nonlinear dynamic systems," in *Proc. Int. Autom. Control Conf. (CACS)*. Chiayi, Taiwan: National Chung Cheng Univ., Nov. 2021.

[22] A. Rospawan, C. C. Tsai, and F. C. Tai, "Adaptive predictive PID control using recurrent fuzzy broad learning system for accurate setpoint tracking of digital nonlinear time-delay dynamic systems," *Int. J. IRobotics*, vol. 5, no. 3, pp. 26–32, Nov. 2022.

[23] 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," M.S. thesis, Dept. Elect. Eng., Nat. Chung Hsing Univ., Taichung, Taiwan, 2020.

[24] A. Rospawan, "Adaptive intelligent SISO and MIMO control methods using output recurrent fuzzy broad learning systems," M.S. thesis, Dept. Elect. Eng., Nat. Chung Hsing Univ., Taichung, Taiwan, 2023.

[25] C. C. Tsai and Y. R. Cheng, "Intelligent PID injection speed and pressure control using ORBLS for hydraulic plunger machine in semiconductor die packaging," in *Proc. Int. Conf. Syst. Sci. Eng.*, Sep. 2020.

[26] 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. Nat. Symp. Syst. Sci. Eng.* Taichung, Taiwan: National Chung Hsing Univ., May 2020.

[27] 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. Int. Conf. Syst. Sci. Eng. (ICSSE)*, Aug. 2021, pp. 482–489, doi: 10.1109/ICSSE52999.2021.9538497.

[28] S. Feng and C. L. P. Chen, "Nonlinear system identification using a simplified fuzzy broad learning system: Stability analysis and a comparative study," *Neurocomputing*, vol. 337, pp. 274–286, Apr. 2019, doi: 10.1016/j.neucom.2019.01.073.

[29] C.-C. Tsai, H.-Y. Chen, S.-C. Chen, F.-C. Tai, and G.-M. Chen, "Adaptive reinforcement learning formation control using ORFBLS for omnidirectional mobile multi-robots," *Int. J. Fuzzy Syst.*, vol. 25, no. 5, pp. 1756–1769, Jul. 2023, doi: 10.1007/s40815-023-01491-4.

[30] C. C. Tsai, H. Y. Chen, C. C. Chan, C. C. Hung, and G. M. Chen, "Intelligent actor-critic learning control for collision-free trajectory tracking of mecanum-wheeled mobile robots," *Int. J. Fuzzy Syst.*, to be published.

[31] N. Farah, M. H. N. Talib, Z. Ibrahim, Q. Abdullah, Ö. Aydogdu, M. Azri, J. B. Mat Lazi, and Z. M. Isa, "Investigation of the computational burden effects of self-tuning fuzzy logic speed controller of induction motor drives with different rules sizes," *IEEE Access*, vol. 9, pp. 155443–155456, 2021, doi: 10.1109/ACCESS.2021.3128351.

[32] H.-G. Han, F.-F. Yang, H.-Y. Yang, and X.-L. Wu, "Type-2 fuzzy broad learning controller for wastewater treatment process," *Neurocomputing*, vol. 459, pp. 188–200, Oct. 2021, doi: 10.1016/j.neucom.2021.06.074.

[33] M. A. George, D. V. Kamat, and C. P. Kurian, "Electronically tunable ACO based fuzzy FOPID controller for effective speed control of electric vehicle," *IEEE Access*, vol. 9, pp. 73392–73412, 2021, doi: 10.1109/ACCESS.2021.3080086.

[34] B. Bian and L. Wang, "A robust fuzzy PD inverse dynamics decoupling control of spherical motion mechanism with fuzzy linear extended state observer," *IEEE Access*, vol. 9, pp. 40140–40154, 2021, doi: 10.1109/ACCESS.2021.3064359.

[35] L. Liu, J. Fei, and C. An, "Adaptive sliding mode long short-term memory fuzzy neural control for harmonic suppression," *IEEE Access*, vol. 9, pp. 69724–69734, 2021, doi: 10.1109/ACCESS.2021.3077646.

[36] L. Yuebo, Y. Haitao, L. Hongyan, and L. Qingxue, "Fuzzy clustering and routing protocol with rules tuned by improved particle swarm optimization for wireless sensor networks," *IEEE Access*, vol. 11, pp. 128784–128800, 2023, doi: 10.1109/ACCESS.2023.3332914.

[37] C. Li, H. Khan, J. Lee, J. Kim, and M. C. Lee, "Fuzzy TSMC-SPO for trajectory tracking of nuclear reactor dismantlement robot manipulator," *IEEE Access*, vol. 11, pp. 38696–38707, 2023, doi: 10.1109/ACCESS.2023.3253213.

[38] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "Self-learning in aerial robotics using type-2 fuzzy systems: Case study in hovering quadrotor flight control," *IEEE Access*, vol. 9, pp. 119520–119532, 2021, doi: 10.1109/ACCESS.2021.3107906.

[39] 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. Int. Conf. Syst. Sci. Eng. (ICSSE)*, May 2022, pp. 11–16, doi: 10.1109/ICSSE55923.2022.9948234.

[40] P. D. Domański, *Control Performance Assessment: Theoretical Analyses and Industrial Practice*. Cham, Switzerland: Springer, Jun. 2022, doi: 10.1007/978-3-030-23593-2.

[41] A. Rospawan, Y. Yang, P.-H. Chen, and C.-C. Tsai, "Study on setpoint tracking performance of the PID SISO and MIMO under noise and disturbance for nonlinear time-delay dynamic systems," *Green Intell. Syst. Appl.*, vol. 2, no. 2, pp. 83–94, Oct. 2022, doi: 10.53623/gisa.v2i2.106.

[42] M.-G. Zhang, Z.-G. Wang, and P. Wang, "Adaptive PID decoupling control based on RBF neural network and its application," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit.*, Nov. 2007, pp. 727–731, doi: 10.1109/icwapr.2007.4420764.

[43] 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.

[44] A. Rospawan, C.-C. Tsai, and C.-C. Hung, "Output recurrent fuzzy neural LSTM-BLS controller for nonlinear digital time-delay dynamic systems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Honolulu, HI, USA, 2023, pp. 3009–3014, doi: 10.1109/SMC53992.2023.10394144.

[45] A. Rospawan, C. C. Tsai, C. C. Hung, and S. S. Chen, "Recurrent fuzzy broad learning controller for MIMO nonlinear digital time-delay dynamic systems," in *Proc. 20th World Congr. Int. Fuzzy Syst. Assoc. (IFSA)*, Aug. 2023.

[46] C. H. Lu, "Adaptive predictive temperature control for an extrusion barrel in plastic injection molding process," M.S. thesis, Dept. Elect. Eng., Nat. Chung Hsing Univ., Taiwan, 2000.

[47] C. H. Lu, "Design and implementation of adaptive predictive controllers for industrial processes," Ph.D. dissertation, Dept. Elect. Eng., Nat. Chung Hsing Univ., Taichung, Taiwan, 2007.

ALI ROSPAWAN received the Diploma 3 degree in electronics engineering from Politeknik Manufaktur Negeri Bangka Belitung, Sungai Liat, Indonesia, in 2015, the B.S. degree in electrical engineering from President University, Cikarang, Indonesia, in 2020, and the M.S. degree from National Chung Hsing University, Taichung, Taiwan, in 2023, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include advanced control methods, intelligent control systems, fuzzy control, evolutionary algorithms, and their application to industrial processes and machinery.

CHING-CHIH TSAI (Fellow, IEEE) received the Diploma degree from the Department of Electrical Engineering, National Taipei Institute of Technology, Taipei, Taiwan, in 1981, the M.S. degree from the Institute of Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1986, and the Ph.D. degree from the Department of Electrical Engineering, Northwestern University, Evanston, IL, USA, in 1991. From 2003 to 2005, he was the Director of the Center of Research Development and Engineering Technology, College of Engineering, National Chung Hsing University (NCHU), Taichung. In 2006, he was the Director of the Center for Advanced Industry Technology and Precision, NCHU. He was the Department Chair of the Department of Electrical Engineering, NCHU, from 2012 to 2014. He is currently a Life Distinguished Professor with the Department of Electrical Engineering, NCHU. He has published and coauthored more than 600 technical articles. His current research interests include fuzzy control systems, advanced control methods, and intelligent learning control systems with their applications to commercial and industrial products, mobile robots, and intelligent machinery. He is a fellow of IET, CACS, RST, and TFSA. Since 2022, he has been a BoG Member and an Associate Vice President of IEEE SMCS. From 2012 to 2016, he was the President of the Chinese Automatic Control Society (CACS), Taiwan, for two-terms. From 2016 to 2019, he was the President of the Robotics Society of Taiwan (RST), for two-terms. From 2019 to 2021, he was the President Elect of IFSA and the IFSA President, from 2021 to 2022.

CHI-CHIH HUNG received the B.S. degree in mechanical engineering from the National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan, in 2003, the M.S. degree in mechanical engineering from I-Shou University, Kaohsiung, in 2006, and the M.S. degree in electrical engineering from National Chung Hsing University, Taichung, Taiwan, in 2014, where he is currently pursuing the Ph.D. degree in electrical engineering. His current research interests include neural networks, fuzzy systems, and intelligent systems.