INTERNATIONAL JOURNAL OF AUTOMOTIVE AND MECHANICAL ENGINEERING
ISSN: 2229-8649 e-ISSN: 2180-1606
VOLUME 22, ISSUE 3, 2025, 12821 - 12836
DOI: https://doi.org/10.15282/ijame.22.3.2025.20.0977

RESEARCH ARTICLE

Adaptive Nonlinear PID Control of DC Motor Position Using a Polynomial Fuzzy Long Short-Term Memory Neural Network

Ali Rospawan[1,2], Clara Lavita Angelina[1,3]*, I Made Andik Setiawan[1], Zanu Saputra[1], Ocsirendi[1], Aan Febriansyah[1], Indra Dwisaputra[1], Robert Napitupulu[4]

[1]Department of Electronics Engineering, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat, 33215, Indonesia
[2]Department of Electrical Engineering, National Chung Hsing University, Taichung, 402202, Taiwan
[3]Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, Yunlin, 64002, Taiwan
[4]Department of Mechanical Engineering, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat, 33215, Indonesia

ABSTRACT – This paper presents an adaptive nonlinear PID control approach enhanced by a polynomial fuzzy long short-term memory (LSTM) neural network-based system identification, addressing the limitations of traditional PID controllers in nonlinear systems. The proposed PFLSTM-ANPID integrates polynomial fuzzy modeling with LSTM-based learning, dynamically adjusting PID gains in real-time without requiring prior knowledge of system dynamics. This innovative adaptive mechanism enhances control precision, disturbance rejection, and real-time adaptability in complex, time-varying environments. Simulation studies on a highly nonlinear system demonstrate that the proposed controller significantly outperforms traditional PID, achieving 80.51% reduction in Root Mean Square Error (RMSE), 81.28% reduction in Integral Absolute Error (IAE), and 85.39% reduction in Integral Time Absolute Error (ITAE). Additionally, it surpasses advanced adaptive controllers, including SISO-CFMFAC variants, attaining the lowest IAE of 98.502. Experimental validation on a DC motor position control task further confirms the controller's superiority, showing a 47.85% improvement in RMSE, 68.91% in IAE and a 78.57% improvement in ITAE compared to traditional PID, and outperforming a fuzzy neural network-based adaptive PID (FNN-APID) controller across all metrics. The proposed approach offers a robust, scalable, and practical solution for industrial applications where system parameters fluctuate significantly over time, representing a meaningful advancement in adaptive control techniques.

ARTICLE HISTORY
Received : 11th Feb. 2025
Revised : 13th May 2025
Accepted : 30th July 2025
Published : 09th Oct. 2025

KEYWORDS
Adaptive PID control
LSTM neural networks
Polynomial fuzzy system
Position control
System Identification

1. INTRODUCTION

Proportional-Integral-Derivative (PID) control is widely used in industrial automation and motion control due to its straightforward form, ease of implementation, and effectiveness in numerous applications. However, traditional PID controllers rely on fixed gain parameters, making them highly sensitive to nonlinearities, variations in system parameters, and dynamic environmental changes. When applied to complex, time-varying systems, this limitation leads to poor tracking performance, increased steady-state errors, and slow response times. Typical tuning approaches such as the Ziegler-Nichols and Cohen-Coon rules often fail to deliver optimal performance under changing operating conditions, as they do not adapt in real-time [1] - [3].

Current adaptive PID control strategies suffer from significant limitations that hinder their practical implementation. Gain scheduling approaches, while simple and industrially reliable, require extensive prior system knowledge and lack real-time flexibility, leading to increased tracking errors when system dynamics deviate from predefined models [4]. Model-based adaptive control offers theoretical convergence guarantees but relies heavily on accurate system models, becoming computationally prohibitive for complex systems, where performance deteriorates by 25-35% when model uncertainties exceed 10% [3]. Heuristic optimization techniques, such as genetic algorithms and particle swarm optimization, provide global optimization capabilities but require more computational intensity than conventional methods and are impractical for real-time applications that require response times of less than 100 ms [5].

Traditional fuzzy logic controllers handle uncertainties through rule-based reasoning without requiring explicit mathematical models; however, their fixed rule bases limit adaptability and hinder performance improvements [6]. Neural network-based approaches excel at learning complex nonlinear functions but lack memory for temporal dependencies, require extensive training data, and suffer from overfitting, with performance degrading 20-30% under unseen operating conditions [4]. These collective limitations reveal three critical gaps in existing adaptive PID control: the inability to simultaneously handle nonlinearities and temporal dependencies, the lack of real-time learning without prior knowledge of the system, and insufficient integration of interpretable fuzzy reasoning with advanced learning capabilities.

To address these challenges and retain the benefits of PID control while improving its adaptability, this paper introduces a Polynomial Fuzzy LSTM neural network-based Adaptive Nonlinear PID Controller, which combines fuzzy

***CORRESPONDING AUTHOR | C. L. Angelina** | ✉ clara_lavita_angelina@polman-babel.ac.id

logic, polynomial function modeling, and deep learning techniques to enhance system adaptability and control precision. Fuzzy logic provides rule-based reasoning to handle uncertainties, while polynomial functions improve nonlinear system approximation, offering a more flexible model than traditional Takagi-Sugeno-Kang (TSK) fuzzy systems dynamics [7], [8]. The LSTM neural network further enhances adaptability by capturing long-term dependencies in system behavior, enabling the controller to learn from past states and adjust PID gains dynamically in real-time dynamics [9]. This integration eliminates the need for prior system knowledge, making the approach suitable for highly nonlinear, time-varying environments.

The inclusion of polynomial functions in the fuzzy system is motivated by the need for higher-order flexibility in modeling nonlinear behaviors. Traditional Takagi-Sugeno-Kang (TSK) fuzzy systems typically employ linear functions in their rule consequents, which may not be sufficient for highly nonlinear systems. By incorporating polynomial functions, the fuzzy system gains a more expressive structure, allowing for better approximation of nonlinearities and improved prediction accuracy in system identification [7]. Furthermore, LSTM networks are introduced to handle time-dependent variations and improve the adaptive learning capability of the system. Unlike conventional neural networks, LSTMs can retain and utilize past information, making them ideal for sequential data processing in control applications [10]. This enables the controller to learn from previous states and adjust its PID gains in real-time, ensuring better stability, robustness, and tracking accuracy in nonlinear control scenarios.

This paper proposes a novel control framework that integrates Polynomial Fuzzy System Identification and LSTM neural network-based learning into an Adaptive Nonlinear PID Controller. The key contributions include: (1) the development of a polynomial fuzzy-based system identification method that achieves over 80% improvement in tracking accuracy compared to traditional approaches, (2) the integration of LSTM networks for temporal pattern learning that reduces integral time absolute error by 85.39%, and (3) a comprehensive stability analysis of the proposed control system. The effectiveness of the proposed approach is validated through both simulation studies on a highly nonlinear system model and real-world experimental validation on a DC motor position control task. The results demonstrate that the proposed method significantly outperforms not only traditional PID control but also advanced adaptive controllers, such as SISO-CFMFAC variants [10], achieving the lowest IAE value of 98.502 compared to 110.393 for SISO-CFMFAC-LSTM. Although the proposed method involves multiple intelligent components, real-time feasibility is maintained by executing all computations on a host computer using MATLAB, with Arduino serving only as a middleware for motor interfacing. This ensures that the control loop runs without delay, even during hardware experiments.

The remainder of this paper is organized as follows: Section 2 reviews related work on adaptive PID control, fuzzy logic-based controllers, and deep learning applications in control systems. Section 3 details the Polynomial Fuzzy LSTM-based system identification model. Section 4 presents the design and stability analysis of the Adaptive Nonlinear PID Controller (ANPID). Section 5 provides simulation studies, while Section 6 discusses the experimental validation of DC motor position control. Finally, Section 7 concludes the work with a summary of the findings and potential future research directions.

2. LITERATURE REVIEW

PID control has been a fundamental technique in industrial automation and control systems for decades. Its simplicity, reliability, and ease of implementation make it a widely adopted approach in various applications, including robotics, process control, and motor drive systems. However, traditional PID controllers are limited by their fixed parameters, making them ineffective in handling system nonlinearities, disturbances, and time-varying dynamics. This limitation becomes particularly critical in precision applications, such as DC motor position control, where even small parameter variations can significantly impact performance. While researchers have explored adaptive and intelligent control techniques to enhance the performance of PID control [11], a significant gap remains between theoretical developments and practical implementation requirements. To improve the adaptability of PID controllers, various adaptive mechanisms have been proposed over the years, ranging from classical model-based techniques to more advanced intelligent control methods. One of the earliest approaches to adaptive control involved gain scheduling, where PID parameters are predefined for different operating conditions and switched based on real-time system measurements. While gain scheduling offers some degree of adaptability, it requires prior knowledge of system dynamics and does not permit continuous learning or real-time tuning [1].

A more sophisticated method involves model-based adaptive PID control, which dynamically adjusts control gains using real-time system identification. These controllers rely on recursive estimation techniques, such as the Least Squares Method (LSM), Recursive Least Squares (RLS), and Kalman Filtering, to continuously update PID parameters based on observed system behavior. Such techniques have been widely used in industrial control systems and robotics [5], [12]. However, they often depend on accurate mathematical models, which can be challenging to obtain in highly nonlinear, uncertain, or time-varying systems [13]. Moreover, in systems with unknown disturbances or external noise, these methods struggle to maintain robust performance, necessitating the integration of more flexible and adaptive learning strategies [14], [15].

To address the limitations of model-based approaches, researchers have explored data-driven and heuristic-based adaptive control strategies. One widely studied approach is the fuzzy logic-based adaptive PID controller, where fuzzy inference systems (FIS) dynamically adjust the PID gains based on system performance indicators such as error

magnitude, rate of change, and acceleration of error [16], [17]. This method does not require an explicit mathematical model, making it particularly effective in uncertain and nonlinear environments. Numerous studies have demonstrated that fuzzy adaptive PID controllers significantly enhance control performance by enabling rule-based decision-making and real-time adaptability. However, conventional fuzzy logic controllers typically rely on fixed rule bases and linear function approximations in their Takagi-Sugeno-Kang (TSK) fuzzy model [3]. These limitations hinder their ability to capture complex, highly nonlinear system interactions, particularly in dynamic environments with fast-changing dynamics.

Fuzzy logic control has been extensively used to address uncertainty and imprecise system modeling. The Takagi-Sugeno-Kang (TSK) fuzzy model is one of the most widely applied fuzzy control approaches, where the consequent part of fuzzy rules is defined by linear functions of input variables [18]. While TSK fuzzy models have demonstrated exemplary performance in various applications, their linear rule consequents are often insufficient for modeling highly nonlinear systems [19]. To overcome this limitation, polynomial fuzzy models have been proposed, where the consequent part for every single fuzzy rule is represented as a polynomial variable rather than a simple linear function. This enhancement allows for greater flexibility in approximating nonlinear dynamics and improves control accuracy in complex systems [20], [21]. Several studies have shown that polynomial fuzzy logic controllers outperform traditional TSK fuzzy controllers, especially in scenarios where system behavior exhibits significant nonlinearity and time-varying characteristics [7], [22].

Neural networks have been widely explored for system identification and control due to their capacity to learn complex input-output relationships. Among deep learning models, Long Short-Term Memory (LSTM) networks have drawn significant attention due to their capacity to handle time-series data and long-term dependencies. Unlike traditional feedforward neural networks, LSTMs use memory cells and gating mechanisms to retain important past information while filtering out irrelevant data [9]. LSTM networks have been successfully applied in predictive modeling, fault diagnosis, and dynamic system control. Recent studies have integrated LSTMs with adaptive control strategies to enhance real-time learning and compensate for system uncertainties. However, most LSTM-based control approaches rely solely on black-box learning, lacking interpretability and direct integration with established control frameworks [10], [23], [24].

Recent comparative studies have shown varying degrees of success in addressing nonlinear control challenges. Traditional adaptive methods, such as SISO-CFMFAC [25], show modest improvements in integral absolute error (24.75% better than traditional PID), while more advanced variants incorporating backpropagation (SISO-CFMFAC-BP) [26] achieve up to a 70.88% improvement. The integration of LSTM networks in SISO-CFMFAC-LSTM further enhances performance, achieving a 79.02% improvement in IAE and an 83.93% reduction in ITAE [10]. However, these approaches still face challenges in balancing computational efficiency with control accuracy, particularly in highly nonlinear systems with rapid parameter variations.

While significant progress has been made in adaptive PID control, fuzzy logic-based systems, and deep learning, existing approaches still face challenges when dealing with highly nonlinear, time-varying, and uncertain systems. Traditional adaptive PID controllers require explicit system models, fuzzy logic controllers struggle with fixed rule bases and linear approximations, and LSTM-based approaches often lack interpretability in control applications. To address these limitations, this paper proposes a Polynomial Fuzzy LSTM neural network-based Adaptive Nonlinear PID Controller, which features the beneficial aspects of fuzzy logic, polynomial function modeling, and LSTM networks. The polynomial fuzzy system enhances nonlinear approximation capabilities, while the LSTM network provides sequential learning and real-time adaptation of PID gains. This hybrid approach demonstrates superior performance across all metrics, achieving 80.51% reduction in RMSE, 81.28% reduction in IAE, and 85.39% reduction in ITAE compared to traditional PID control, while maintaining practical implementation feasibility. The effectiveness of this approach is demonstrated through comprehensive simulation studies and real-world experimental validation on DC motor position control.

3. POLYNOMIAL FUZZY LSTM NEURAL NETWORK

3.1 Polynomial Fuzzy Long Short-Term Memory Neural Network System Identification

System identification is a crucial step in control design, particularly for nonlinear dynamic systems, where traditional linear models often struggle to capture complex system behaviors. To address this challenge, the Polynomial Fuzzy Model (PFM) is employed, integrating Takagi-Sugeno-Kang (TSK) fuzzy logic with polynomial functions. This hybrid approach provides a flexible and adaptive framework for accurately modeling nonlinear system dynamics. Additionally, a Long Short-Term Memory (LSTM) network is incorporated to enhance the algorithm's capability to capture sequential dependencies and long-term temporal patterns in the system's behavior. A general nonlinear system can be expressed as:

$$y(k+1) = f\big(y(k), y(k-1), \cdots, y\big(k-n_y\big), u(k), u(k-1), \cdots, u(k-n_u)\big) \tag{1}$$

where $y(k)$ represents the system's actual output at time step k while $u(k)$ denotes the system input. The parameters n_y and n_u define the proportion of past output and input values used for prediction. The function $f(\cdot)$ is an unknown nonlinear mapping function that is approximated using polynomial fuzzy logic.

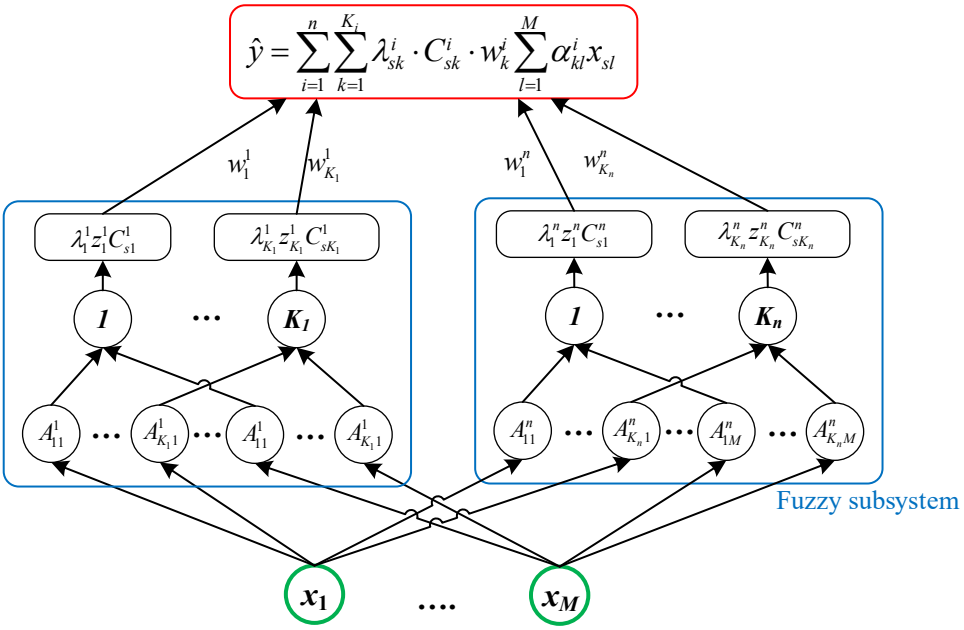

Figure 1. Polynomial fuzzy LSTM neural network architecture

The PFM extends the TSK fuzzy model by utilizing polynomial functions in the consequent part of fuzzy rules instead of conventional linear functions. This enhancement allows for more flexible approximations of complex system behaviors. Following the neural network architecture of the proposed polynomial fuzzy LSTM neural network, depicted in Figure 1, the system consists of three main layers, where the LSTM is part of the fuzzy subsystem in the second layer. Furthermore, a typical fuzzy rule in the polynomial fuzzy model is formulated as [18]:

$$\begin{aligned} &\text{IF } x_{s1} \text{ IS } A_{k1}^i \text{ AND } x_{s2} \text{ IS } A_{k2}^i, \cdots, x_{sM} \text{ IS } A_{kM}^i \\ &\text{THEN } z_{sk}^i = f_k^i(x_{s1}, x_{s2}, \cdots, x_{sM}), k = 1,2, \cdots, K_i \end{aligned} \tag{2}$$

To approximate $f(\cdot)$, the system is divided into fuzzy regions, where each region is represented by a local polynomial model. The identification system uses a first-order polynomial TSK fuzzy model, expressed as:

$$z_{sk}^i = f_k^i(x_{s1}, x_{s2}, \cdots, x_{sM}) = \sum_{l=1}^{M} \alpha_{kl}^i x_{sl} \tag{3}$$

where z_{sk}^i is the local model output in the i-th fuzzy subsystem for rule k. x_{sM} represents the input variables $\big(y(k), y(k-1), \cdots, y\big(k-n_y\big), u(k), u(k-1), \cdots, u(k-n_u)\big)$ and the parameter α_{kl}^i is a coefficient defining the local model structure.

To determine the degree of activation of each fuzzy rule, the Gaussian membership function and its firing strength are obtained in the following:

$$\mu_{kt}^i = exp\left(-\left(\frac{x - c_{kt}^i}{\sigma_{kt}^i}\right)^2\right); \quad \tau_{sk}^i = \prod_{t=1}^{M} \mu_{kt}^i(x_{st}) \tag{4}$$

where c_{kt}^i and σ_{kt}^i represent the width and center.

To further enhance the algorithm's predictive capability and capacity to capture temporal dependencies, while strengthening its robustness against noise, an LSTM network is integrated into the identification framework [27]. The inner working LSTM cell is governed by the following update equations [9], [28]:

i) Forget Gate: controls how much past information should be retained:

$$f_{sk}^i(k) = \sigma\big(w_{fk} \cdot \phi_{sk}^i(k-1) + r_{fk} \cdot \tau_{sk}^i + b_{fk}\big) \tag{5}$$

ii) Input Gate: determines how new information is incorporated:

$$u_{sk}^i(k) = \sigma\big(w_{uk} \cdot \phi_{sk}^i(k-1) + r_{uk} \cdot \tau_{sk}^i + b_{uk}\big) \tag{6}$$

iii) Candidate State Update: computes potential new memory content:

$$a_{sk}^i(k) = tanh\big(w_{ak} \cdot \phi_{sk}^i(k-1) + r_{ak} \cdot \tau_{sk}^i + b_{ak}\big) \tag{7}$$

iv) Cell State Update: updates the LSTM memory:

$$c_{sk}^{i}(k) = c_{sk}^{i}(k-1) \circ f_{sk}^{i}(k) + u_{sk}^{i}(k) \circ a_{sk}^{i}(k) \tag{8}$$

v) Output Gate: generates the final output of the memory cell:

$$o_{sk}^{i}(k) = \sigma\left(w_{ok} \cdot \phi_{sk}^{i}(k-1) + r_{ok} \cdot \tau_{sk}^{i} + b_{ok}\right) \tag{9}$$

vi) Final LSTM Output Computation:

$$\phi_{sk}^{i}(k) = o_{sk}^{i}(k) \circ tanh\left(c_{sk}^{i}(k)\right) \tag{10}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ and $tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$ are the sigmoid and the hyperbolic tangent activation functions [23], [24].

The weighted fuzzy neural LSTM firing strength is then computed as:

$$\lambda_{sk}^{i} = \frac{\phi_{sk}^{i}(k)}{\sum_{k=1}^{K_i} \phi_{sk}^{i}(k)} \tag{11}$$

Then, to enhance the efficiency of the fuzzy rule-based model, fuzzy C-means clustering is applied, which determines the degree of rule membership using [20], [22]:

$$C_{sk}^{i} = \frac{1}{\sum_{j=1}^{K_i}\left(\frac{||x_s - c_{kt}^{i}||^2}{||x_s - c_{kj}^{i}||^2}\right)} \tag{12}$$

where C_{sk}^{i} represents the fuzzy cluster weight for rule k, ensuring that similar data points belong to the same fuzzy region.

The final system identification output is computed as follows:

$$\hat{y} = \sum_{i=1}^{n}\sum_{k=1}^{K_i} \lambda_{sk}^{i} \cdot C_{sk}^{i} \cdot w_{k}^{i} \sum_{l=1}^{M} \alpha_{kl}^{i} x_{sl} \tag{13}$$

where w_k^i reflects the rule weight for rule k in subsystem i and the term $\sum_{l=1}^{M} \alpha_{kl}^{i} x_{sl}$ ensures flexibility in approximating system behavior.

3.2 Gradient Descent-Based Weights Update Algorithm

To optimize the system's response, an error objective function is employed to minimize the discrepancy between the predicted or system identification output and actual system outputs. The objective function is defined as follows:

$$E(k) = \frac{1}{2}(\hat{y}(k) - y(k))^2 \tag{14}$$

where $\hat{y}(k)$ denotes the predicted system output, and $y(k)$ is the actual system response at sampling instance k. The optimization rules aim to adjust the weight parameters of the polynomial fuzzy system to minimize this error, thereby enhancing the system's accuracy.

Then, the weight parameters of the polynomial fuzzy LSTM neural network model are updated using the gradient descent algorithm, which iteratively adjusts the weights based on the error gradient:

$$\boldsymbol{P}(k+1) = \boldsymbol{P}(k) + \Delta\boldsymbol{P}(k) = \boldsymbol{P}(k) - \eta_i(k)\frac{\partial E(k)}{\partial \boldsymbol{P}(\boldsymbol{k})} \tag{15}$$

where $\boldsymbol{P}$ represents any of the fuzzy neural network weights $\left(w_k^i, c_{kt}^i, \text{and } \sigma_{kt}^i\right)$ and LSTM weights, including; w_{fk}, r_{fk}, b_{fk} of the forget Gate, w_{uk}, r_{uk}, b_{uk} of the input gate, w_{ak}, r_{ak}, b_{ak} of the cell gate and w_{fk}, r_{fk}, b_{fk} of the LSTM output gate.

Then, the weight parameters of the polynomial fuzzy model are updated using the gradient descent optimization property, which iteratively adjusts the weights based on the error gradient. The weight update equations are given by:

$$w_k^i(k+1) = w_k^i(k) - \eta_i(k)\frac{\partial E(k)}{\partial w_k^i(k)} \tag{16}$$

$$c_{kt}^i(k+1) = c_{kt}^i(k) - \eta_i(k)\frac{\partial E(k)}{\partial c_{kt}^i(k)} \tag{17}$$

$$\sigma_{kt}^i(k+1) = \sigma_{kt}^i(k) - \eta_i(k)\frac{\partial E(k)}{\partial \sigma_{kt}^i(k)} \tag{18}$$

where $\eta_i(k)$ is the gradient scaling factor, controlling the step size of the weight updates. Since the gradient of the error function with respect to the predicted output is: $\partial E/\partial \hat{y} = \hat{y} - y = \hat{e}$, then the partial derivatives of the error function with respect to the fuzzy system parameters are computed as:

$$\frac{\partial E(k)}{\partial w_k^i(k)} = \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial w_k^i(k)} = (\hat{y}(k) - y(k)) \cdot \lambda_k^i \left(\sum_{l=1}^{M} \alpha_{kl}^i x_{sl} \right) \tag{19}$$

$$\frac{\partial E(k)}{\partial c_{kl}^i(k)} = \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial c_{kl}^i(k)} = (\hat{y}(k) - y(k)) \cdot \sum_{i=1}^{n} \sum_{k=1}^{K_i} \lambda_{sk}^i \left(\sum_{l=1}^{M} \alpha_{kl}^i x_{sl} \right) \cdot \frac{2\left(x - c_{kt}^i\right)}{\left(\sigma_{kt}^i\right)^2} \tag{20}$$

$$\frac{\partial E}{\partial \sigma_{kl}^i} = \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \sigma_{kl}^i(k)} = (\hat{y}(k) - y(k)) \cdot \sum_{i=1}^{n} \sum_{k=1}^{K_i} \lambda_{sk}^i \left(\sum_{l=1}^{M} \alpha_{kl}^i x_{sl} \right) \cdot \frac{2\left(x - c_{kt}^i\right)^2}{\left(\sigma_{kt}^i\right)^3} \tag{21}$$

Furthermore, to ensure consistency in LSTM weight updates, the weights are rearranged based on their respective roles within the LSTM structure. The gradients of the error function with respect to the forget gate weights are first computed as [23], [24]:

$$\begin{aligned} \frac{\partial E(k)}{\partial w_{fk}(k)} &= \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^i(k)} \cdot \frac{\partial \phi_{sk}^i(k)}{\partial f_{sk}^i(k)} \cdot \frac{\partial f_{sk}^i(k)}{\partial w_{fk}(k)} \\ &= \left(\hat{y}(k) - y(k)\right) \cdot w_k^i(k)\, o_{sk}^i(k) \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) c_{sk}^i(k-1) f_{sk}^i \left(1 - f_{sk}^i\right) \tau_{sk}^i \end{aligned} \tag{22}$$

$$\begin{aligned} \frac{\partial E(k)}{\partial r_{fk}(k)} &= \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^i} \cdot \frac{\partial \phi_{sk}^i(k)}{\partial f_{sk}^i(k)} \cdot \frac{\partial f_{sk}^i(k)}{\partial r_{fk}(k)} \\ &= \left(\hat{y}(k) - y(k)\right) w_k^i\, o_{sk}^i \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) c_{sk}^i(k-1) f_{sk}^i \left(1 - f_{sk}^i\right) \phi_{sk}^i(k-1) \end{aligned} \tag{23}$$

$$\begin{aligned} \frac{\partial E(k)}{\partial b_{fk}(k)} &= \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^i(k)} \cdot \frac{\partial \phi_{sk}^i(k)}{\partial f_{sk}^i(k)} \cdot \frac{\partial f_{sk}^i(k)}{\partial b_{fk}(k)} \\ &= (\hat{y}(k) - y(k)) w_k^i(k)\, o_{sk}^i(k) \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) c_{sk}^i(k-1) f_{sk}^i \left(1 - f_{sk}^i\right) \end{aligned} \tag{24}$$

Similarly, the gradients of the error function with respect to the input gate weights are computed as:

$$\begin{aligned} \frac{\partial E(k)}{\partial w_{uk}(k)} &= \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^i(k)} \cdot \frac{\partial \phi_{sk}^i(k)}{\partial u_{sk}^i(k)} \cdot \frac{\partial u_{sk}^i(k)}{\partial w_{uk}(k)} \\ &= (\hat{y}(k) - y(k)) w_k^i(k)\, o_{sk}^i(k) \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) a_{sk}^i\, u_{sk}^i \left(1 - u_{sk}^i\right) \tau_{sk}^i \end{aligned} \tag{25}$$

$$\begin{aligned} \frac{\partial E(k)}{\partial r_{uk}(k)} &= \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^i(k)} \cdot \frac{\partial \phi_{sk}^i(k)}{\partial u_{sk}^i(k)} \cdot \frac{\partial u_{sk}^i(k)}{\partial r_{uk}(k)} \\ &= (\hat{y}(k) - y(k)) w_k^i(k)\, o_{sk}^i(k) \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) a_{sk}^i\, u_{sk}^i \left(1 - u_{sk}^i\right) \phi_{sk}^i(k-1) \end{aligned} \tag{26}$$

$$\begin{aligned} \frac{\partial E(k)}{\partial b_{uk}(k)} &= \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^i(k)} \cdot \frac{\partial \phi_{sk}^i(k)}{\partial u_{sk}^i(k)} \cdot \frac{\partial u_{sk}^i(k)}{\partial b_{uk}(k)} \\ &= (\hat{y}(k) - y(k)) w_k^i(k)\, o_{sk}^i(k) \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) a_{sk}^i\, u_{sk}^i \left(1 - u_{sk}^i\right) \end{aligned} \tag{27}$$

For the cell gate weights, the gradients are determined as follows:

$$\begin{aligned} \frac{\partial E(k)}{\partial w_{ak}(k)} &= \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^i(k)} \cdot \frac{\partial \phi_{sk}^i(k)}{\partial a_{sk}^i(k)} \cdot \frac{\partial a_{sk}^i(k)}{\partial w_{ak}(k)} \\ &= \left(\hat{y}(k) - y(k)\right) w_k^i(k)\, o_{sk}^i(k) \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) u_{sk}^i \left(1 - (u_{sk}^i)^2\right) \tau_{sk}^i \end{aligned} \tag{28}$$

$$\begin{aligned} \frac{\partial E(k)}{\partial r_{ak}(k)} &= \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^i(k)} \cdot \frac{\partial \phi_{sk}^i(k)}{\partial a_{sk}^i(k)} \cdot \frac{\partial a_{sk}^i(k)}{\partial r_{ak}(k)} \\ &= (\hat{y}(k) - y(k)) w_k^i(k)\, o_{sk}^i(k) \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) u_{sk}^i \left(1 - (u_{sk}^i)^2\right) \phi_{sk}^i(k-1) \end{aligned} \tag{29}$$

$$\frac{\partial E(k)}{\partial b_{ak}(k)} = \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^{i}(k)} \cdot \frac{\partial \phi_{sk}^{i}(k)}{\partial a_{sk}^{i}(k)} \cdot \frac{\partial a_{sk}^{i}(k)}{\partial b_{ak}(k)} = \big(\hat{y}(k) - y(k)\big) w_k^i(k)\, o_{sk}^i(k) \left(1 - \left(tanh\left(c_{sk}^i(k)\right)\right)^2\right) u_{sk}^i \big(1 - (u_{sk}^i)^2\big) \tag{30}$$

Lastly, the gradients of the error function with respect to the output gate weights are given by:

$$\frac{\partial E(k)}{\partial w_{ok}(k)} = \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^{i}(k)} \cdot \frac{\partial \phi_{sk}^{i}(k)}{\partial o_{sk}^{i}(k)} \cdot \frac{\partial o_{sk}^{i}(k)}{\partial w_{ok}(k)} = \big(\hat{y}(k) - y(k)\big) w_k^i(k) \left(tanh\left(c_{sk}^i(k)\right)\right) o_{sk}^i \big(1 - o_{sk}^i\big) \tau_{sk}^i \tag{31}$$

$$\frac{\partial E(k)}{\partial r_{ok}(k)} = \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^{i}(k)} \cdot \frac{\partial \phi_{sk}^{i}(k)}{\partial o_{sk}^{i}(k)} \cdot \frac{\partial o_{sk}^{i}(k)}{\partial r_{ok}(k)} = \big(\hat{y}(k) - y(k)\big) w_k^i(k) \left(tanh\left(c_{sk}^i(k)\right)\right) o_{sk}^i \big(1 - o_{sk}^i\big) \phi_{sk}^i(k-1) \tag{32}$$

$$\frac{\partial E(k)}{\partial b_{ok}(k)} = \frac{\partial E(k)}{\partial \hat{y}(k)} \cdot \frac{\partial \hat{y}(k)}{\partial \phi_{sk}^{i}(k)} \cdot \frac{\partial \phi_{sk}^{i}(k)}{\partial o_{sk}^{i}(k)} \cdot \frac{\partial o_{sk}^{i}(k)}{\partial b_{ok}(k)} = \big(\hat{y}(k) - y(k)\big)\, w_k^i(k) \left(tanh\left(c_{sk}^i(k)\right)\right) o_{sk}^i \big(1 - o_{sk}^i\big) \tag{33}$$

where w, r, b represent the input weights, recurrent weights, and the bias parameters [28], [29].

3.3 Identification System Convergence Evaluation

Ensuring the convergence of the Polynomial Fuzzy LSTM Neural Network requires a bounded learning rate. The system convergence is analyzed using Lipschitz continuity, where the Hessian matrix of the system is approximated to determine the appropriate learning rate range [6], [30]. The Hessian matrix, which describes the second-order derivatives of the error function with reference to the system parameters, is given by:

$$H(k) = \nabla^2 E(\boldsymbol{P}(k)) = \frac{\partial^2 E(k)}{\partial \boldsymbol{P^2}(k)} \tag{34}$$

where $\boldsymbol{P}$ represents the set of all polynomial fuzzy LSTM neural network weight parameters. Since the Hessian provides insight into the curvature of the objective function, its largest eigenvalue determines the worst-case sensitivity of the system's weight updates. Then, the Lipschitz constant $\boldsymbol{L}$ is defined as:

$$L(k) = \max eig\ (H(k)) \tag{35}$$

where $eig\ (H)$ represents the eigenvalues of the Hessian matrix.

Finally, for stable weight updates, the gradient descent learning rate must satisfy the following conditions:

$$0 < \eta_i(k) < \frac{2}{L(k)} \tag{36}$$

where in practical implementation, the learning rate is dynamically adjusted as: $\eta_i(k) = min\big(\eta_i(k), (2/L(k))\big)$. This ensures bounded weight updates and convergence of the polynomial fuzzy LSTM neural network system identification process.

To prove the convergence of the gradient descent-based optimization, we consider the Taylor series expansion of the error function $E(P)$ around the current parameter set $P(k)$:

$$E\big(P(k+1)\big) = E\big(P(k)\big) + \nabla E\big(P(k)\big)^T \Delta P + \frac{1}{2} \Delta P^T H \Delta P + O(\|\Delta P\|^3) \tag{37}$$

substituting the gradient descent update rule $\Delta P = -\eta_i(k)\ \nabla E(P(k))$ yields:

$$E\big(P(k+1)\big) = E\big(P(k)\big) - \eta_i(k)\ \big\|\nabla E\big(P(k)\big)\big\|^2 + \frac{1}{2} \eta_i^2(k) \nabla E\big(P(k)\big)^T H \nabla E(P(k)) \tag{38}$$

For convergence, the error must decrease with each iteration, which requires $E\big(P(k+1)\big) < E\big(P(k)\big)$. This holds if:

$$-\eta_i(k)\ \big\|\nabla E\big(P(k)\big)\big\|^2 + \frac{1}{2} \eta_i^2(k) \nabla E\big(P(k)\big)^T H \nabla E\big(P(k)\big) < 0 \tag{39}$$

Factoring out η_i:

$$\eta_i(k) \left(- \big\|\nabla E\big(P(k)\big)\big\|^2 + \frac{1}{2} \eta_i(k) \nabla E\big(P(k)\big)^T H \nabla E\big(P(k)\big) \right) < 0 \tag{40}$$

Since H_i is positive semi-definite, its eigenvalues are non-negative, and we obtain:

$$\frac{1}{2}\eta_i(k)L(k)\left\|\nabla E\left(P(k)\right)\right\|^2 < \left\|\nabla E\left(P(k)\right)\right\|^2 \tag{41}$$

which simplifies to:

$$\eta_i(k) < \frac{2}{L(k)} \tag{42}$$

Thus, setting η_i within this bound ensures a monotonic decrease in the error function, proving convergence of the learning process [6].

4. ADAPTIVE NONLINEAR PID CONTROL DESIGN

This section presents the Adaptive Nonlinear PID (ANPID) Control approach, which leverages the polynomial fuzzy LSTM neural network-based system identification model for real-time parameter adaptation. Traditional PID controllers with fixed gains often struggle in nonlinear environments, leading to inadequate control performance. In practice, the PID controller is first tuned using conventional methods to ensure baseline stability. To address the limitations of fixed-gain designs, an adaptive mechanism is then introduced to enable real-time tuning of PID parameters based on system dynamics. The polynomial fuzzy LSTM model subsequently refines these gains online to enhance tracking accuracy and improve disturbance rejection.

4.1 ANPID Control Design

As depicted in Figure 2, the closed-loop control structure of the proposed ANPID consists of two main parts: the ANPID control block and the polynomial fuzzy LSTM neural network block. The ANPID block generates the controller output as the controlled plant input, while the neural network block generates the change rate of PID gains as an adjustment to continuously update the PID gains over time. The standard PID control law for computing the control input $u(k)$ at time step k is given by:

$$u(k) = k_p \cdot e(k) + k_i \cdot \big(e(k) - e(k-1)\big) + k_d \cdot (e(k) - 2(e(k-1)) + e(k-2)) \tag{43}$$

where $e(k) = r(k) - y(k)$ is the control error, representing the deviation across the reference signal $r(k)$ and system output $y(k)$. k_p, k_i and k_d are the proportional, integral, and derivative gains, respectively [31].

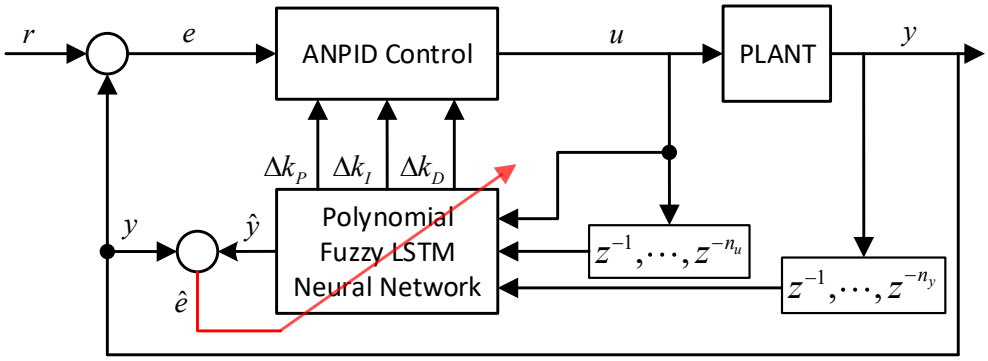

Figure 2. Polynomial fuzzy LSTM neural network-based ANPID closed-loop control structure

Since fixed PID parameters perform poorly in nonlinear conditions, an adaptive mechanism is introduced, where PID gains are dynamically updated based on the identified polynomial fuzzy model. The adaptation mechanism is derived from an error-based cost function:

$$J(k) = \frac{1}{2}(r(k) - \hat{y}(k))^2 \tag{44}$$

where $\hat{y}(k)$ is the predicted system response. Using gradient-based adaptation, the PID gains evolve as follows [6]:

$$k_P(k+1) = k_P(k) + \Delta k_P(k) = k_P(k) - \eta_c(k)\frac{\partial J(k)}{\partial k_P(k)} \tag{45}$$

$$k_I(k+1) = k_I(k) + \Delta k_I(k) = k_I(k) - \eta_c(k)\frac{\partial J(k)}{\partial k_I(k)} \tag{46}$$

$$k_D(k+1) = k_D(k) + \Delta k_D(k) = k_D(k) - \eta_c(k)\frac{\partial J(k)}{\partial k_D(k)} \tag{47}$$

where η_c is the adaptive learning rate that ensures stable parameter updates.

The gradients of the cost function relative to the PID gains are computed as:

$$\frac{\partial J(k)}{\partial k_P(k)} = \big(r(k) - \hat{y}(k)\big) \cdot G_u(k) \cdot \big(e(k) - e(k-1)\big) \tag{48}$$

$$\frac{\partial J(k)}{\partial k_I(k)} = \big(r(k) - \hat{y}(k)\big) \cdot G_u(k) \cdot e(k) \tag{49}$$

$$\frac{\partial J(k)}{\partial k_D(k)} = (r(k) - \hat{y}(k)) \cdot G_u(k) \cdot (e(k) - 2(e(k-1)) + e(k-2)) \tag{50}$$

where $G_u(k)$ represents the system sensitivity to current control input, which is typically close to 1 in steady-state conditions. These adaptation rules enable the controller to adjust its gains dynamically, improving responsiveness and robustness to system variations [32].

4.2 Control System Stability Analysis

The stability of the adaptive nonlinear PID control system is analyzed using a Hessian-based approach, similar to the one presented in Section 3.3. The Hessian matrix of the cost function is defined as:

$$H(k) = \nabla^2 E(\boldsymbol{P_c}) = \frac{\partial^2 E(k)}{\partial \boldsymbol{P_c^2(k)}} \tag{51}$$

where $\boldsymbol{P_C}$ is the parameter vector containing the adaptive PID gains k_p, k_i and k_d.

To ensure stability, the learning rate η_c must satisfy the Lipschitz condition:

$$0 < \eta_c(k) < \frac{2}{L_c(k)} \tag{52}$$

where L_c is the maximum eigenvalue of the Hessian matrix, given by:

$$L_c(k) = \max eig\ (H(k)) = \max eig\ \left(\frac{\partial^2 J(k)}{\partial k_p^2(k)}, \frac{\partial^2 J(k)}{\partial k_i^2(k)}, \frac{\partial^2 J(k)}{\partial k_d^2(k)}\right) \tag{53}$$

To ensure stable adaptation, the learning rate is dynamically updated as $\eta_c(k) = min\big(\eta_c(k), (2/L_c(k))\big)$. By bounding the learning rate, the adaptive PID controller avoids excessive gain fluctuations, ensuring that parameter updates remain stable and the control system remains well-behaved.

4.3 REAL-TIME CONTROL ALGORITHM

This pseudocode (Algorithm 1) outlines the control algorithm for DC motor position control using an adaptive nonlinear PID controller with Polynomial Fuzzy System Identification and LSTMN-based learning.

Algorithm 1: ANPID control-based polynomial fuzzy LSTM neural network

Input: $\big(y(k), y(k-1), \cdots, y\big(k-n_y\big), u(k), u(k-1), \cdots, u(k-n_u)\big)$ and $(e(k), \Delta e(k), \Delta^2 e(k))$.

Output: Controller output u.

1. Parameter initialization.
2. **FOR** $k = 1$ **TO** T:
3. Read motor position $y(k)$.
4. Compute error $e(k)$, $\Delta e(k)$, and $\Delta^2 e(k)$.
5. Construct input vector: $X = \big[y(k), y(k-1), \cdots, y\big(k-n_y\big), u(k), u(k-1), \cdots, u(k-n_u)\big]$.
6. Compute control signal $u(k)$.
7. Compute PID gradient updates: $\frac{\partial J(k)}{\partial k_P(k)}, \frac{\partial J(k)}{\partial k_I(k)}$, and $\frac{\partial J(k)}{\partial k_D(k)}$.
8. Compute the adaptive learning rate of η_c.
9. Update PID gains: k_p, k_i and k_d.
10. Compute polynomial fuzzy LSTM neural network gradient updates: $\frac{\partial E}{\partial \boldsymbol{P}}$.
11. Update the polynomial fuzzy LSTM neural network $\boldsymbol{P}$.
12. Update the learning rate η_i.
13. Apply Control Input by sending $u(k)$ to Arduino and the motor driver.
14. **END**

5. SIMULATION STUDY OF NONLINEAR SYSTEM MODEL

To evaluate the effectiveness of the proposed control framework, we conduct a simulation study using a highly nonlinear system model adapted from [10]. The system dynamics are governed by the following equation:

$$y(k+1) = \frac{2.5y(k)y(k-1)}{1+y^2(k)+y^2(k-1)} + 1.2u(k) + 0.09u(k)u(k-1) + 1.6u(k-2) + 0.7\,sin\Big(0.5\big(y(k)+y(k-1)\big)\Big) \times cos\big(0.5(y(k)+y(k-1))\big) \tag{54}$$

This model introduces considerable nonlinear complexity due to its combination of multiplicative interactions, trigonometric expressions, and interdependencies between past system outputs and control inputs. These characteristics make it an ideal benchmark for assessing the adaptive capabilities of the proposed Polynomial Fuzzy LSTM Neural Network-based adaptive nonlinear PID controller.

To evaluate the tracking effectiveness of the controller under dynamic conditions, a reference trajectory was defined as follows:

$$r(k+1) = 5\,sin(k\pi/50) + 2\,cos(k\pi/20) \tag{55}$$

where this trajectory introduces continuous variations and rapid setpoint transitions, posing a challenge for controllers that rely on fixed or slowly adapting parameters.

The simulation parameter settings are divided into three key components. One of the initial PID gains is $K_p = 0.05$, $k_i = 0.2$, and $k_d = 0.1$ with a controller learning rate $\eta_c = 0.01$. The second is for the system identification parameters, where the number of past outputs used $n_y = 3$ and past input used $n_u = 5$. The number of fuzzy rules $K = 4$, fuzzy subsystem $n = 4$, system identification learning rate $\eta_i = 0.01$. Lastly, the initial parameters for the LSTM network were initialized with small random values for the input and recurrent weights, within the range [0, 1]. At the same time, the bias vector was assigned a small positive value of 0.1 to ensure proper activation of the LSTM gates.

Table 1. Controller performance comparison for nonlinear system simulation

Controller	RMSE	IAE	ITAE
Traditional PID	3.812	526.086	75831.938
SISO-CFMFAC	4.402	395.882	20877.901
SISO-CFMFAC-BP	0.995	153.182	13314.346
SISO-CFMFAC-LSTM	0.699	110.393	12187.146
Polynomial Fuzzy LSTMNN AN PID	0.743	98.502	11077.235

Table 2. Percentage of improvement compared to traditional PID controller

Controller	RMSE	IAE	ITAE
SISO-CFMFAC	-15.48%	24.75%	72.47%
SISO-CFMFAC-BP	73.90%	70.88%	82.44%
SISO-CFMFAC-LSTM	81.66%	79.02%	83.93%
Polynomial Fuzzy LSTMNN AN PID	80.51%	81.28%	85.39%

To quantitatively assess the tracking accuracy and control effectiveness, three widely recognized performance indices were employed: the root mean square error (RMSE), the integral absolute error (IAE), and the integral time-weighted absolute error (ITAE). These metrics provide a comprehensive evaluation of both short-term and long-term performance. The detailed formulations of these indices can be found in [10], [33], [34]. The proposed method's performance was also directly compared with conventional and advanced control approaches, as documented in [10]. The results, summarized in Table 1 for the overall performance comparison of five traditional and recent proposed approaches and Table 2 for the improvement percentage compared to traditional PID control, demonstrate that the proposed controller significantly outperforms conventional PID [33], [35], CFMFAC [25], and CFMFAC-BP [26] in terms of RMSE, IAE, and ITAE. However, when compared to the CFMFAC-LSTM controller [10], the proposed method exhibits better performance in terms of IAE and ITAE but shows slightly higher RMSE. This suggests that while the proposed method achieves superior long-term stability and cumulative error reduction, further tuning may be required to enhance short-term precision.

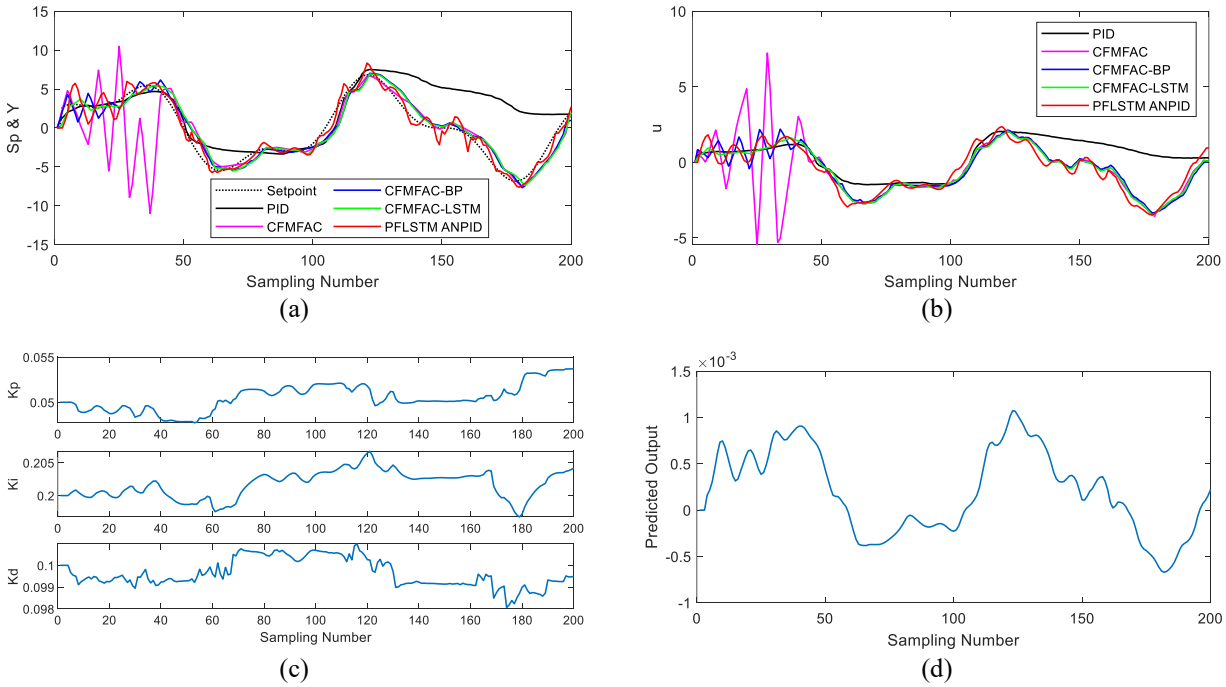

Figure 3. Simulation result of the nonlinear system model setpoint tracking control. (a) setpoint tracking output comparative results. (b) controller output comparative results. (c) PID gains evolution. (d) system predicted output

The nonlinear setpoint tracking response, illustrated in Figure 3(a), highlights the effectiveness of the proposed approach in handling complex nonlinear behaviors. Unlike the traditional PID controller, which struggles to maintain stability and deviates significantly from the reference trajectory, the proposed method demonstrates consistent tracking performance. Despite a relatively higher overshoot compared to some alternative approaches, the proposed controller maintains reliable tracking even in rapidly changing conditions. Figure 3(b) presents the controller output, which exhibits noticeable fluctuations in the initial 40 sampling instances before stabilizing. This behavior suggests an adaptation phase in which the controller fine-tunes its parameters before achieving steady-state control. Figure 3(c) illustrates the time evolution of the PID gains, showcasing how the controller dynamically adjusts its parameters to accommodate nonlinear system behavior. Lastly, Figure 3(d) plots the system identification predicted output, confirming the accuracy of the Polynomial Fuzzy LSTM model in capturing and approximating nonlinear system dynamics.

Overall, the findings presented in Figure 3, Table 1, and Table 2 validate the superior performance of the proposed control strategy. Even in the presence of significant nonlinearities, the method demonstrates robust tracking accuracy and effective adaptation. These results confirm that integrating polynomial fuzzy modeling with LSTM-based learning provides a powerful solution for nonlinear system control, surpassing conventional approaches in handling complex system dynamics.

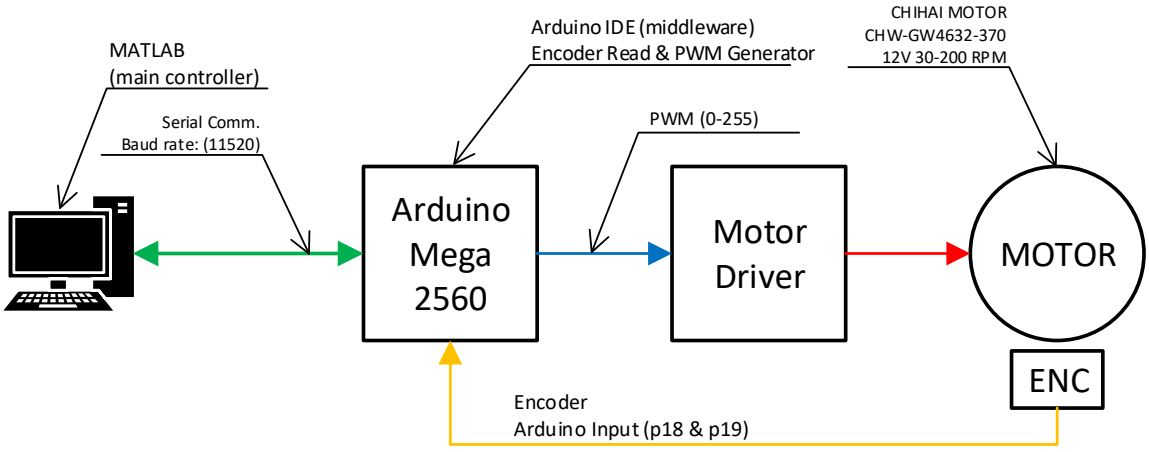

Figure 4. Experimental devices integration chart

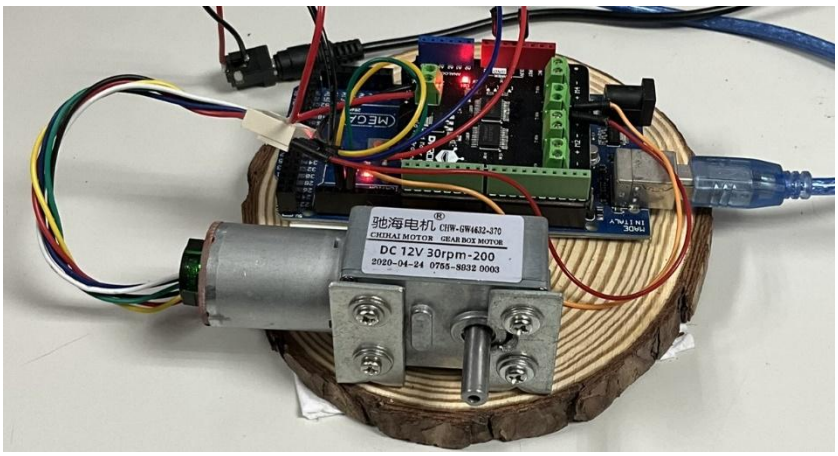

Figure 5. Actual experimental DC motor positioning control setup

6. DC MOTOR POSITION CONTROL: AN EXPERIMENTAL STUDY

This section describes the experimental setup and procedures used to validate the effectiveness of the proposed adaptive nonlinear PID controller, which is enhanced by a polynomial fuzzy LSTM neural network-based system identification for DC motor position control. The primary objective of this experiment is to evaluate the controller's performance in real-world conditions, specifically its ability to handle nonlinearities and transient state variations.

The experimental setup follows the device integration framework shown in Figure 4, while the actual testbed configuration is depicted in Figure 5. The DC motor used in this experiment is equipped with an encoder sensor for position feedback, and its operation is managed through a motor control driver, which is mounted on an intermediary middleware device (Arduino Mega 2560). This middleware device performs three key functions: (1) Sensor Data Acquisition, where it reads position data from the encoder and transmits it to the main controller (computer). (2) PWM Actuator Interface that receives control signals in the form of PWM duty cycles from the main controller and applies them to the motor. Lastly, (3) Communication Bridge utilized a serial communication (USB/UART) at a baud rate of 115200 to enable real-time and fast data exchange between the computer and the DC motor system.

The main controller is a standard personal computer running MATLAB, which serves as the central processing unit responsible for executing the control algorithm. It processes sensor input, implements the polynomial fuzzy LSTM-based system identification model, computes the adaptive nonlinear PID control output, and sends the corresponding control signal to the DC motor in real-time. It is essential to note that the computational load, encompassing neural network inference and fuzzy rule evaluations, is entirely handled within MATLAB on the PC. As long as the PC meets MATLAB's minimum system requirements, the system can operate in real-time without noticeable latency. This design choice ensures that the proposed framework remains practical and computationally feasible in prototyping and experimental phases, even if embedded deployment is a potential future direction.

For experimental evaluation, the system parameters were carefully configured. The initial PID gains were set as $K_p = 1.5$, $k_i = 0.6$, and $k_d = 0.8$, with a controller learning rate of $\eta_c = 0.001$. The system identification settings included a past output window size of $n_y = 3$ and a past input window size of $n_u = 3$, ensuring a balance between historical dependency and real-time adaptability. The fuzzy inference system utilized 10 fuzzy rules ($K = 10$) and 10 fuzzy subsystems ($n = 10$). The system identification learning rate was set to $\eta_i = 0.001$. The LSTM network was initialized with random small values for the input and recurrent weights, within the range [0, 1], while the bias vector was initialized with a small positive value of 0.1, ensuring proper activation of the LSTM gates during the early training phase.

The experimental results, illustrated in Figure 6 and quantified in Table 3 and 4, provide detailed insights into the control performance of the proposed PFLSTM-ANPID controller compared to traditional PID [33] and FNN-APID approaches [36], [37]. Figure 6(a) presents the setpoint tracking performance of the DC motor position control for all three controllers, demonstrating the superior ability of the proposed PFLSTM-ANPID approach to follow the desired reference trajectory with minimal deviation. As shown in Table 3, our approach achieves an RMSE of 0.0688, outperforming both traditional PID (RMSE = 0.13193) and FNN-APID (RMSE = 0.0774). Figure 6(b) displays the controller outputs of the proposed PFLSTM-ANPID. Figure 6(c) illustrates the evolution of PID gains over time in our adaptive approach, showing that the gains undergo rapid changes between time steps 0 and 30 before gradually stabilizing between approximately 100 and 360 time steps. These adaptive variations in PID gains are crucial in handling the harsh transient-state response observed at the beginning of the experiment.

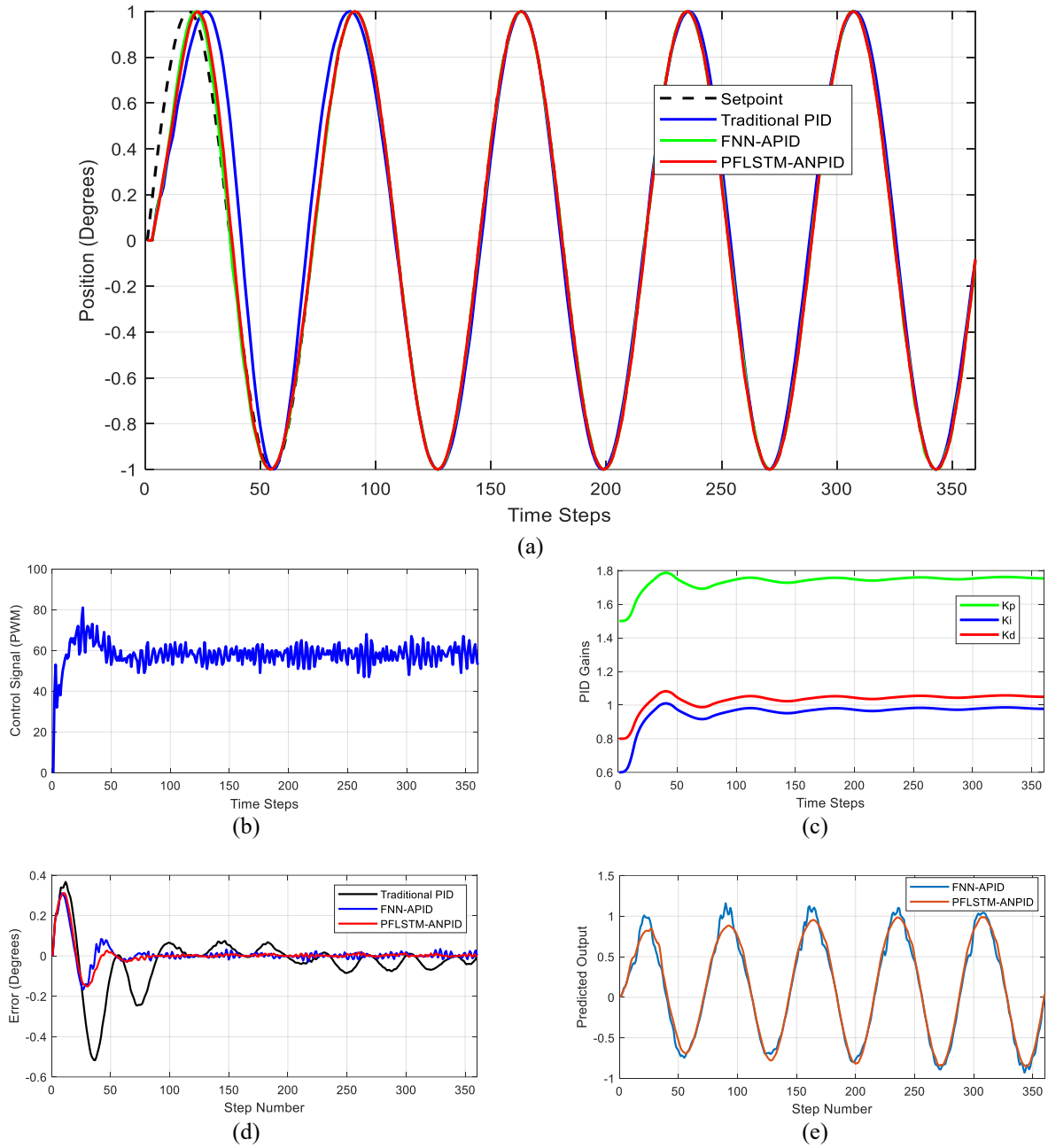

Figure 6. Experimental result for DC motor position control; (a) position tracking output comparison; (b) PFLSTM-ANPID controller output; (c) PID gains evolutions of PFLSTM-ANPID; (d) position error comparison; (e) system predicted output comparison

Table 3. Controller performance comparison for DC motor position control

Controller	RMSE	ISE	IAE	ITAE
Traditional PID	0.13193	6.2661	27.3000	2678.90
FNN-APID	0.07740	1.8868	8.7071	639.04
PFLSTM-ANPID	0.06880	1.5729	8.4865	574.13

Table 4. Percentage of improvement compared to traditional PID controller

Controller	RMSE	ISE	IAE	ITAE
FNN-APID	41.333%	69.889%	68.106%	76.145%
PFLSTM-ANPID	47.851%	74.898%	68.914%	78.568%

To further analyze tracking accuracy, Figure 6(d) plots the error between the desired position and the actual system output for all three controllers. The PFLSTM-ANPID consistently maintains lower error values throughout both transient and steady-state phases, confirming that our controller successfully minimizes tracking errors over time. This is

quantitatively supported by the IAE and ITAE metrics in Table 3, where PFLSTM-ANPID achieves the lowest values (IAE = 8.4865, ITAE = 574.13) compared to traditional PID (IAE = 27.3, ITAE = 2678.9) and FNN-APID (IAE = 8.7071, ITAE = 639.04). Finally, Figure 6(e) illustrates the output of the polynomial fuzzy LSTM-based system identification model compared to FNN-APID, demonstrating a stable and highly accurate prediction model that plays a key role in dynamically adjusting the PID gains to enhance system performance. Table 4 quantifies the percentage improvements of both adaptive controllers over traditional PID, with PFLSTM-ANPID achieving significant improvements across all metrics: 47.851% in RMSE, 74.898% in ISE, 68.914% in IAE, and 78.568% in ITAE. Notably, PFLSTM-ANPID consistently outperforms FNN-APID across all metrics, with additional improvements of 6.518% in RMSE, 5.009% in ISE, 0.808% in IAE, and 2.423% in ITAE.

Experimental findings confirm that the proposed approach achieves superior control performance, effectively mitigating significant transient responses while maintaining excellent steady-state precision. This enhanced performance is attributed to three key factors:

vii) The strong learning capability of the polynomial fuzzy LSTM neural network, which more accurately captures complex nonlinear dynamics compared to traditional fuzzy neural networks.

viii) The adaptive mechanism dynamically adjusts the PID gains in real-time based on predicted system behavior.

ix) The polynomial fuzzy modeling approach is better suited to handle nonlinearities in the system.

The system offers three key advantages: simplicity, as it does not require extensive tuning or complex configurations; computational efficiency, since the Arduino Mega 2560, combined with MATLAB running on a standard computer, manages the entire control process; and adaptability, as it can rapidly respond to environmental variations without requiring prior knowledge of system dynamics or an offline training phase. These characteristics make the proposed approach a robust and scalable solution for nonlinear control applications in industrial settings where system parameters frequently fluctuate.

7. CONCLUSION

This paper presented a novel adaptive nonlinear PID control approach enhanced by a polynomial fuzzy LSTM neural network-based system identification to address the fundamental challenges of nonlinear system control. The proposed PFLSTM-ANPID framework uniquely integrates polynomial fuzzy modeling, LSTM-based sequential learning, and real-time adaptive control, enabling precise and robust system response across varying dynamic conditions without requiring prior knowledge of system dynamics.

Simulation studies validated the effectiveness of the proposed controller in handling a highly nonlinear system model, demonstrating remarkable improvements over traditional PID and other advanced control strategies. Specifically, the proposed method achieved an 80.51% reduction in RMSE, 81.28% reduction in IAE, and 85.39% reduction in ITAE compared to traditional PID control. Furthermore, our approach outperformed advanced adaptive controllers, including SISO-CFMFAC and its variants, achieving the lowest IAE value of 98.502, compared to 110.393 for SISO-CFMFAC-LSTM and 395.882 for basic SISO-CFMFAC.

Experimental validation on DC motor position control further confirmed the superiority of our approach, demonstrating significant performance improvements over both traditional PID and FNN-based adaptive PID controllers. As quantified in Tables 3 and 4, the PFLSTM-ANPID controller achieved improvements of 47.85% in RMSE, 74.89% in ISE, 68.91% in IAE, and 78.57% in ITAE compared to traditional PID control. Notably, it also consistently outperformed the FNN-APID controller across all metrics, with additional improvements of 6.52% in RMSE, 5.01% in ISE, 0.81% in IAE, and 2.42% in ITAE. These results demonstrate that our controller effectively mitigates transient-state fluctuations while adapting to system nonlinearities in real-time.

Key advantages of the proposed PFLSTM-ANPID approach include: (1) Superior error reduction capabilities, demonstrated by significant improvements in RMSE over both traditional PID and other adaptive control methods. (2) Enhanced temporal performance, shown by substantial reductions in ITAE, indicating better long-term stability and faster convergence. (3) Improved overall accuracy with significant reductions in IAE and ISE, confirming better tracking performance throughout operation. (4) Real-time adaptation capability that maintains computational efficiency while outperforming other adaptive methods, as demonstrated by the experimental implementation on standard hardware.

Unlike traditional controllers with fixed parameters and previous adaptive approaches with limited learning capabilities, the self-tuning mechanism of our proposed method allows the system to dynamically adjust its control strategy based on the polynomial fuzzy LSTM prediction model, enhancing both tracking accuracy and stability across varying operating conditions. The comprehensive comparative analysis of both traditional PID and advanced adaptive controllers (FNN-APID and SISO-CFMFAC variants) demonstrates that our approach achieves better performance across all metrics while maintaining simplicity of implementation. This study confirms that the novel integration of polynomial fuzzy logic, LSTM networks, and adaptive PID control provides a robust, efficient, and scalable solution for nonlinear control applications in industrial settings where system parameters frequently fluctuate. Future research will focus on enhancing computational efficiency, extending the method to multivariable nonlinear systems, and enabling real-time deployment on embedded platforms through hardware optimization and lightweight system identification

implementation. Additionally, we will explore benchmarking against reinforcement learning and advanced MPC controllers to further assess the competitiveness of our approach in complex control scenarios.

ACKNOWLEDGEMENTS

The authors would like to express sincere gratitude to Politeknik Manufaktur Negeri Bangka Belitung, National Yunlin University of Science and Technology, and National Chung Hsing University for providing the resources and support necessary to conduct this research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of the paper.

AUTHORS CONTRIBUTION

A. Rospawan contributed to the methodology, data analysis and original draft preparation.

C. L. Angelina assisted with the methodology and contributed to manuscript review and editing.

I. M. A. Setiawan was responsible for formal analysis and project supervision.

Z. Saputra handled data analysis and assisted with validation.

Ocsirendi contributed to software implementation and technical validation.

A. Febriansyah supported project supervision and conducted a literature review.

I. Dwisaputra was involved in data acquisition and project administration.

R. Napitupulu contributed to final manuscript validation.

REFERENCES

[1] 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.

[2] S. I. Haris, F. Ahmad, M. H. C. Hassan, A. K. M. Yamin, and N. R. M. Nuri, "Self tuning PID control of antilock braking system using electronic wedge brake," *International Journal of Automotive and Mechanical Engineering*, vol. 18, no. 4, pp. 9333–9348, 2021.

[3] P. Eze, D. O. Njoku, O. C. Nwokonkwo, C. G. Onukwugha, J. N. Odii, and J. E. Jibiri, "Wheel slip equilibrium point model reference adaptive control based PID controller for antilock braking system: a new approach," *International Journal of Automotive and Mechanical Engineering*, vol. 21, no. 3, pp. 11581–11595, 2024.

[4] O. Dogru, K. V. Velswamy, F. Ibrahim,Y. Wu, A. S. Sundaramoorthy, B. Huang, et al., "Reinforcement learning approach to autonomous PID tuning," *Computer and Chemical Engineering*, vol. 161, p. 107760, May 2022.

[5] A. Taeib, A. Ltaeif, and A. Chaari, "A PSO approach for optimum design of multivariable PID controller for nonlinear systems," *arXiv preprint arXiv:1306.6194*, 2022.

[6] A. Rospawan, C. C. Tsai, and C. C. Hung, "Output recurrent fuzzy broad learning systems for adaptive MIMO PID control: theory, simulations, and application," *IEEE Access*, vol. 12, pp. 19388–19404, 2024.

[7] A. Rospawan and C. C. Tsai, "Recurrent polynomial-based FBLS for adaptive predictive PID control of nonlinear discrete-time systems: comparative studies on control performance and time complexity," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 541–546, 2024.

[8] Y. Li and L. Li, "Observer-based H∞ fault-tolerant control for 2-D polynomial fuzzy large-scale systems with fault-detection," *International Journal of Control, Automation and Systems*, vol. 23, pp. 1484–1496, 2025.

[9] A. Rospawan, C. C. Tsai, and C. C. Hung, "Output recurrent fuzzy neural LSTM-BLS controller for nonlinear digital time-delay dynamic systems," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3009–3014, 2023.

[10] Y. Yang, C. Chen, and J. Lu, "Parameter self-tuning of SISO compact-form model-free adaptive controller based on long short-term memory neural network," *IEEE Access*, vol. 8, pp. 151926–151937, 2020.

[11] 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," *International Conference on System Science and Engineering (ICSSE)*, pp. 011–016, 2022.

[12] 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," *International Journal Fuzzy Systems*, vol. 25, no. 5, pp. 1756–1769, 2023.

[13] J. Meng, W. Sun, Q. F. Pan, and M. X. Ruan, "Research and application of improved particle swarm fuzzy PID algorithm based on self-disturbance rejection in temperature control system of plastic extruder," *IEEE Access*, vol. 12, pp. 41620–41630, 2024.

[14] G. Bujgoi and D. Sendrescu, "Tuning of PID Controllers Using Reinforcement Learning for Nonlinear System Control," *Processes*, vol. 13, no. 3, p. 735, 2025.

[15] T. -C. Lee, C. L. Angelina, P. Kongkam, H. -P. Wang, R. Rerknimitr, M. -L. Han, et al., "Deep-learning-enabled computer-aided diagnosis in the classification of pancreatic cystic lesions on confocal laser endomicroscopy," *Diagnostics*, vol. 13, no. 7, p. 1289, 2023.

[16] 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," *Chinese Journal of Electrical Engineering*, vol. 7, no. 2, pp. 70–82, 2021.

[17] H. Nagai and H. Seki, "Deep modular fuzzy inference model," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1860–1865, 2023.

[18] 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, 2019.

[19] N. Elias and N. M. Yahya, "Simulation study for controlling direct current motor position utilising fuzzy logic controller," *International Journal of Automotive and Mechanical Engineering*, vol. 15, no. 4, 2018.

[20] Z. Zhang, H. D. Yang, K. Xu, and C. Zhu, "Fuzzy C-means model based multimodeling for nonlinear distributed parameter systems," *International Conference on Control Science and Systems Engineering (ICCSSE)*, pp. 420–426, 2023.

[21] W. Huang, S.-K. Oh, and W. Pedrycz, "Fuzzy wavelet polynomial neural networks: analysis and design," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1329–1341, 2017.

[22] J. Huang, C.-M. Vong, G. Wang, W. Qian, Y. Zhou, and C. L. P. Chen, "Joint label enhancement and label distribution learning via stacked graph regularization-based polynomial fuzzy broad learning system," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 9, pp. 3290–3304, 2023.

[23] 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.

[24] M. Liu, W. Zhou, and Z. Xu, "Hesitant fuzzy long short-term memory network and its application in the intelligent building selection," *IEEE Transactions on Fuzzy Systems*, vol. 32, no. 5, pp. 2590–2602, 2024.

[25] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2173–2188, 2011.

[26] C. Chen and J. Lu, "Design of self-tuning SISO partial-form model-free adaptive controller for vapor-compression refrigeration system," *IEEE Access*, vol. 7, pp. 125771–125782, 2019.

[27] C. C. Tsai, C. C. Hung, C.-F. Mao, H. S. Wu, and C. H. Chen, "Fuzzy neural LSTM-RBLS for fractional-order PID sliding-mode motion control of autonomous mobile robots with four ISID wheels," *International Journal of Fuzzy Systems*, 2024.

[28] W. Yuan and L. Chao, "Online evolving interval type-2 intuitionistic fuzzy LSTM-neural networks for regression problems," *IEEE Access*, vol. 7, pp. 35544–35555, 2019.

[29] H. Wang, C. Luo, and X. Wang, "Synchronization and identification of nonlinear systems by using a novel self-evolving interval type-2 fuzzy LSTM-neural network," *Engineering Applications of Artificial Intelligence*, vol. 81, pp. 79–93, 2019.

[30] Y. Guo, Z. Hou, S. Liu, and S. Jin, "Data-driven model-free adaptive predictive control for a class of MIMO nonlinear discrete-time systems with stability analysis," *IEEE Access*, vol. 7, pp. 102852–102866, 2019.

[31] 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," *International Journal of iRobotics*, vol. 5, no. 3, pp. 26–32, 2022.

[32] 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," *International Journal of Fuzzy Systems*, vol. 19, no. 6, pp. 1718–1730, 2017.

[33] 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 Intelligent Systems and Applications*, vol. 2, no. 2, p. 2, 2022.

[34] A. Rospawan, C. C. Tsai, and C. C. Hung, "Intelligent MIMO ORFBLS-based setpoint tracking control with its application to temperature control of an industrial extrusion barrel," *International Journal of Fuzzy Systems*, vol. 27, pp. 774–790, 2025.

[35] S. Ekinci, D. Izci, V. Gider, M. Bajaj, V. Blazek, and L. Prokop, "Quadratic interpolation optimization-based 2DoF-PID controller design for highly nonlinear continuous stirred-tank heater process," *Scientific Reports*, vol. 15, p. 16324, 2025.

[36] C. Ben Jabeur and H. Seddik, "Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot," *Asian Journal of Control*, vol. 23, no. 1, pp. 23–41, 2021.

[37] D. H. Pham, C. M. Lin, V. N. Giap, V. P. Vu, and H. Y. Cho, "Design of missile guidance law using Takagi-Sugeno-Kang (TSK) elliptic type-2 fuzzy brain imitated neural networks," *IEEE Access*, vol. 11, pp. 53687–53702, 2023.