

Article

A Data-Driven Predictive Fuzzy Adaptive Control for Nonlinearly Parameterized Systems with Unknown Disturbance

Hongyun Yue , Dongpeng Xue, Yi Zhao and Jiaqi Wang

School of Science, Xi'an University of Architecture and Technology, Xi'an 710055, China; soledad2021@xauat.edu.cn (D.X.); zhaoyi6706@xauat.edu.cn (Y.Z.); ajlifitimeyz@xauat.edu.cn (J.W.)

* Correspondence: yuehongyun0417@xauat.edu.cn

Abstract

Problem: Controlling nonlinearly parameterized systems with unknown disturbances remains challenging because classical adaptive approaches rely on separation-of-variables and reparameterization techniques, leading to increased parameter dimensions, conservative stability bounds, and implementation complexity. **Objective:** This paper develops a data-driven predictive fuzzy adaptive control (DD-PFAC) framework that eliminates the need for separation techniques while achieving superior tracking performance and formally certified stability. **Novelty:** The key innovation is a two-layer architecture. *Layer 1* provides direct fuzzy approximation of composite nonlinear functions (system dynamics plus disturbance bound) without parameter reparameterization, reducing parameter complexity from $O(qn)$ to $O(nN)$. *Layer 2* employs Hankel matrix-based predictive optimization to adaptively tune both control gains $c_i(k)$ and adaptation rates $\gamma_i(k)$ online using 80–150 recent input–output samples. **Methodology:** A Lyapunov function augmented with a prediction-error term is used to prove uniform ultimate boundedness of all closed-loop signals. A projection-based recursive least-squares algorithm updates the gain parameters online while guaranteeing $c_i(k) \geq c_{\min} > 0$ at all times. **Results:** Comparative simulations demonstrate 31.4% reduction in integral square error, 27.8% reduction in mean absolute error, and 37.4% reduction in steady-state error versus traditional adaptive fuzzy control. A four-group ablation study confirms that adaptive gain scheduling contributes 27.7% and predictive compensation contributes 6.5% to the total MAE improvement. Robustness tests validate consistent 28–32% performance advantage across sinusoidal, pulse, step, and large-disturbance scenarios.

Keywords: nonlinearly parameterized systems; adaptive fuzzy control; data-driven predictive control; backstepping; Hankel matrix; Lyapunov stability

MSC: 93C42; 93C40; 93D21



Academic Editor: Hsien-Chung Wu

Received: 14 March 2026

Revised: 2 April 2026

Accepted: 8 April 2026

Published: 11 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

When systems contain unknown nonlinear functions, fuzzy logic systems (FLS) or neural networks can approximate such functions, enabling the design of adaptive controllers with provable stability properties. Adaptive fuzzy control has been extensively applied to diverse nonlinear system classes over the past three decades.

Wang [1] established the foundational universal approximation theorem for fuzzy logic systems, proving that any continuous function on a compact set can be approximated

to arbitrary precision. Chen and Jiao [2] developed adaptive tracking methods for periodically time-varying and nonlinearly parameterized systems using multilayer neural networks. Liu et al. [3] extended adaptive neural control to discrete-time nonlinear systems with output feedback, while Liu et al. [4] further designed reduced-order observer-based controllers for uncertain nonlinear SISO systems. Wang and Chen [5] addressed adaptive fuzzy tracking for strict-feedback nonlinear systems with unknown virtual control coefficients. Wang and Chen [6] further studied perturbed strict-feedback systems with time delay under the fuzzy adaptive framework. Yue and Li [7] proposed observer-based fuzzy adaptive schemes for perturbed systems with unknown control directions. Tong and Li [8] developed observer-based fuzzy adaptive control for strict-feedback nonlinear systems. Li and Yue [9] addressed adaptive fuzzy tracking for stochastic nonlinear systems with unknown time-varying delays. Tong and Li [10] extended these results to MIMO nonlinear systems, while Li and Yue [11] considered MIMO systems with both input and state delays. More recently, Liu et al. [12] proposed a finite-time adaptive fuzzy backstepping scheme for nonlinear systems with full-state constraints, and Li et al. [13] developed event-triggered adaptive fuzzy control that reduces communication overhead while preserving stability guarantees. Zhang et al. [14] further extended data-driven adaptive control to systems with quantized inputs and stochastic disturbances. Li and Tong [15] proposed adaptive backstepping control with full state triggering for strict-feedback nonlinear systems, achieving improved tracking accuracy through state-triggered sampling strategies.

Nonlinearly parameterized systems arise widely in robotics, aerospace, and process control. Lin and Qian [16] proposed the seminal separation-of-variables and reparameterization approach for designing adaptive controllers for such systems. Wu et al. [17] applied neural networks to adaptive tracking of nonlinearly parameterized systems with unknown input nonlinearities. Yue and Yu [18] developed adaptive fuzzy tracking for stochastic nonlinearly parameterized systems with distributed input delay. Long et al. [19] studied switched adaptive control of switched nonlinearly parameterized systems with unstable subsystems. Gao and Yuan [20,21] investigated adaptive stabilization for stochastic nonholonomic systems with nonlinear parameterization. More recently, Benslimane [22] extended the nonlinearly parameterised framework to non-strict-feedback systems with full-state constraints via adaptive iterative learning control, confirming that the reparameterisation approach remains the dominant but increasingly burdensome paradigm for this problem class. However, the separation-of-variables principle and reparameterization technique significantly increase controller complexity, requiring $O(qn)$ additional parameters for an n -dimensional system with a q -dimensional unknown parameter. *How to design a tracking controller for nonlinearly parameterized systems using fuzzy logic systems without the separation-of-variables technique is one of the central focuses of this paper.*

In recent years, data-driven control methods have attracted widespread attention because they do not require accurate mathematical models. In the context of this paper, *data-driven* refers to the paradigm in which the controller uses only measured input–output data $\{u(k), y(k)\}$ from past operation—rather than a first-principles model—to predict future system behavior and to update control parameters online. Hou and Jin [23] introduced model-free adaptive control (MFAC) using pseudo-partial derivatives estimated from input–output data. De Persis and Tesi [24] established rigorous formulas for data-driven control addressing stabilization, optimality, and robustness. Willems et al. [25] provided fundamental results on persistence of excitation essential for data informativity. Coulson et al. [26] proposed data-enabled predictive control (DeePC) using Hankel matrices with finite-sample guarantees. Huang et al. [27] strengthened DeePC with tractable robustness formulations and closed-loop performance guarantees under noisy data. Tanaskovic et al. [28] developed a data-driven control scheme for nonlinear systems via an online direct approach.

Berberich and Allgöwer [29] provide a comprehensive recent overview of systems-theoretic guarantees in data-driven MPC, identifying the extension to nonlinear systems as a key open challenge that the present work addresses. In particular, data-driven model predictive control (DD-MPC) constructs prediction models from historical input–output data and optimizes control strategies online, showing superior performance in handling uncertainty and constraint satisfaction.

1.1. Comparison with Reinforcement Learning Approaches

Recent advances in reinforcement learning (RL) have demonstrated impressive results in robotics and adaptive control [30–32]. However, fundamental differences exist between RL-based and the proposed data-driven predictive approach.

Sample efficiency. Deep RL algorithms (e.g., DDPG [31], SAC [32]) typically require millions of environment interactions for convergence—approximately 10^6 samples for a typical drone control task. In contrast, DD-PFAC uses 80–150 recent samples for short-term prediction, representing a $10^4 \times$ advantage in sample efficiency. This dramatic difference arises because RL learns a global policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ valid across the entire state space, while DD-PFAC leverages the known control-affine structure and predicts only local behavior near the current trajectory.

Safety guarantees. RL lacks formal stability guarantees during exploration; agents may violate safety constraints while learning optimal policies. DD-PFAC maintains Lyapunov-based stability certificates through the base adaptive layer at all times. Even if data-driven predictions are inaccurate, the tracking error remains bounded by Theorem 1.

Applicability comparison. Table 1 summarizes the key differences. RL excels when no structural knowledge is available and extensive offline simulation is feasible. DD-PFAC is preferred when the control-affine structure is known, sample efficiency is critical, and safety must be maintained throughout operation.

Table 1. Comparison of DD-PFAC and Reinforcement Learning Methods.

Aspect	Reinforcement Learning	DD-PFAC (Ours)
System knowledge	None required	Control-affine structure
Training data	$\sim 10^6$ samples	80–150 samples
Stability guarantee	None during training	Lyapunov-based (always)
Computation	Heavy offline training	Light online optimization
Best application	Fully unknown systems	Structured systems

1.2. Main Contributions

The main contributions of this paper are as follows.

1. Methodological innovation. A two-layer architecture is proposed that eliminates the traditional separation-of-variables requirement. Layer 1 directly approximates composite functions $h_i(x_i, \theta, d) = f_i(x_i, \theta) + \bar{d}_i^2(x_i)$ using FLS without reparameterization, reducing parameter complexity from $O(qn)$ to $O(nN)$. Layer 2 employs Hankel matrix-based data-driven prediction using 80–150 samples to optimize control gains $c_i(k)$ and adaptation rates $\gamma_i(k)$ online.

2. Theoretical rigor. A Lyapunov function augmented with a prediction-error term is constructed (Equation (22)). Theorem 1 proves uniform ultimate boundedness of all closed-loop signals regardless of prediction quality. The projection mechanism in Algorithm 1 guarantees $c_i(k) \in [c_{\min}, c_{\max}]$ at all times, ensuring the stability condition $\lambda_{\min}(C(k)) \geq c_{\min} > 0$ holds uniformly.

3. Comprehensive experimental validation. Four ablation groups, four disturbance scenarios, three sensitivity analyses, and four demanding test cases are conducted. The

results confirm 31.4% ISE reduction and consistent 28–37% improvements in all error metrics compared to traditional adaptive fuzzy control (AFC).

Algorithm 1 Data-Driven Control Parameter Online Optimization

```

1: Initialize:  $\mathbf{c}(0) = \mathbf{c}_0, \gamma(0) = \gamma_0, \mathbf{P}(0) = P_0 \mathbf{I}, k = 0$ 
2: repeat
3:   Measure  $x(k), y(k)$ ; update data window  $\mathcal{D}$ 
4:   Construct Hankel matrices  $\mathcal{H}_u, \mathcal{H}_y$  per (5)
5:   Compute predicted outputs  $\hat{y}(k+j|k), j = 1, \dots, N_p$ , using (7)
6:   Compute prediction error  $e_p(k) = y(k) - \hat{y}(k|k-1)$ 
7:   Construct regression vector  $\boldsymbol{\phi}(k) = \left[ \frac{\partial J_p}{\partial c_1}, \dots, \frac{\partial J_p}{\partial c_n} \right]^T$ 
8:   Compute gain matrix  $\mathbf{K}(k) = \frac{\mathbf{P}(k-1)\boldsymbol{\phi}(k)}{\lambda + \boldsymbol{\phi}^T(k)\mathbf{P}(k-1)\boldsymbol{\phi}(k)}$ 
9:   Update parameters:  $\mathbf{c}(k) = \mathbf{c}(k-1) - \mathbf{K}(k)\nabla_{\mathbf{c}} J_p(k)$ 
10:  Project to feasible region:  $\mathbf{c}(k) = \text{proj}[\mathbf{c}(k), \mathbf{c}_{\min}, \mathbf{c}_{\max}] \triangleright \text{ensures } c_i(k) \geq c_{\min} > 0$ 
11:  Update covariance:  $\mathbf{P}(k) = \frac{1}{\lambda} [\mathbf{P}(k-1) - \mathbf{K}(k)\boldsymbol{\phi}^T(k)\mathbf{P}(k-1)]$ 
12:  Compute  $u(k)$  via (11) and (12)
13:   $k \leftarrow k + 1$ 
14: until convergence or maximum iterations reached
  
```

2. Problem Description and Preliminaries

2.1. System Description

Consider the following nonlinearly parameterized system:

$$\begin{cases} \dot{x}_i = x_{i+1} + f_i(x_i, \theta) + d_i(t), & i = 1, \dots, n-1 \\ \dot{x}_n = u + f_n(x_n, \theta) + d_n(t) \\ y = x_1 \end{cases} \quad (1)$$

where $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ is the state vector, $x_i = [x_1, \dots, x_i]^T, y \in \mathbb{R}$ and $u \in \mathbb{R}$ are the output and control input, $\theta \in \mathbb{R}^q$ is an unknown parameter vector, $f_i(\cdot) : \mathbb{R}^i \times \mathbb{R}^q \rightarrow \mathbb{R}$ is an unknown smooth nonlinear function, and $d_i(t)$ is the unknown external disturbance.

The objective is to design a DD-PFAC controller for system (1) such that y tracks a reference signal y_r , the tracking error converges to a small neighborhood of the origin, and all closed-loop signals remain bounded.

Assumption 1. $y_r \in \Omega_{y_r}$ where $\Omega_{y_r} \subset \mathbb{R}^n$ is a known compact set; the reference trajectory $y_r(t)$ and its derivatives up to order n are continuous and bounded.

Assumption 2. For $i = 1, \dots, n$, there exists a positive unknown smooth function $\bar{d}_i(\cdot)$ such that $|d_i(t)| \leq \bar{d}_i(x_i)$.

Assumption 3. The historical input–output data $\{u(k-j), y(k-j), j = 1, \dots, L\}$ are measurable and stored, where $L \geq 2(T_h + T_p)n$ is the data window length.

2.2. Fuzzy Logic System

The following FLS rules are used:

$$R^l: \text{if } x_1 \text{ is } F_1^l \text{ and } \dots \text{ and } x_n \text{ is } F_n^l, \text{ then } y \text{ is } G^l, \quad l = 1, \dots, N$$

Using center-average defuzzification, product inference, singleton fuzzification, and Gaussian membership functions

$\mu_{F_i^l}(x_i) = \exp\left[-\left(\frac{x_i - c_i^l}{b_i^l}\right)^2\right]$, the FLS output is:

$$y(x) = \frac{\sum_{l=1}^N \bar{y}^l \prod_{i=1}^n \mu_{F_i^l}(x_i)}{\sum_{l=1}^N \prod_{i=1}^n \mu_{F_i^l}(x_i)} = \theta^T \zeta(x) \quad (2)$$

where $\theta = [\bar{y}^1, \dots, \bar{y}^N]^T$ and the fuzzy basis function vector $\zeta(x) = [\zeta_1(x), \dots, \zeta_N(x)]^T$ with

$$\zeta_l(x) = \frac{\prod_{i=1}^n \mu_{F_i^l}(x_i)}{\sum_{l=1}^N \prod_{i=1}^n \mu_{F_i^l}(x_i)}. \quad (3)$$

Lemma 1 ([1]). Let $f(x)$ be continuous on a compact set $U \subset \mathbb{R}^n$. Then for any $\varepsilon > 0$ there exists an FLS (2) such that $\sup_{x \in U} |f(x) - \theta^T \zeta(x)| \leq \varepsilon$.

Remark 1 (Rationale for Choosing FLS over Alternatives). FLS is chosen over neural networks, radial basis functions, and polynomial approximators for three reasons. **(1) Universal approximation with interpretable structure:** FLS satisfies Lemma 1 while each rule corresponds to an identifiable local operating region, enabling domain-knowledge-based initialization. **(2) Computational efficiency:** FLS evaluation requires $O(N \cdot n)$ operations. For the second-order system studied here with 7 membership functions per variable, 49 rules are evaluable in <0.1 ms on standard hardware, versus $O(10^3)$ operations for a comparable two-hidden-layer neural network. **(3) Smooth gradient properties:** Gaussian membership functions provide continuously differentiable outputs essential for Lyapunov analysis; their gradients appear explicitly in adaptive laws and have known bounds facilitating stability proofs.

2.3. Data-Driven Predictive Model

To fully exploit historical operational data, this paper introduces a Hankel-matrix-based predictive model. The construction proceeds in four explicit steps.

Step 1: Data collection. Over window $[k - L, k - 1]$, collect:

$$\mathcal{D} = \{u(k - L), \dots, u(k - 1), y(k - L), \dots, y(k - 1)\}. \quad (4)$$

Equation (4) defines the data set used in all subsequent Hankel matrix constructions. The input sequence $\mathbf{u}_{\text{past}} \in \mathbb{R}^L$ and output sequence $\mathbf{y}_{\text{past}} \in \mathbb{R}^L$ store the $2L$ recent scalar measurements.

Step 2: Hankel matrix construction. With Hankel depth T_h and prediction horizon T_p , define:

$$\mathcal{H}_u = \begin{bmatrix} u(k - L) & u(k - L + 1) & \cdots & u(k - T_h - T_p) \\ u(k - L + 1) & u(k - L + 2) & \cdots & u(k - T_h - T_p + 1) \\ \vdots & \vdots & \ddots & \vdots \\ u(k - L + T_h + T_p - 1) & \cdots & \cdots & u(k - 1) \end{bmatrix} \quad (5)$$

with dimensions $\mathcal{H}_u \in \mathbb{R}^{(T_h + T_p) \times N}$, $N = L - T_h - T_p + 1$. Similarly construct $\mathcal{H}_y$. Typical values: $T_h = 10$, $T_p = 10$, $L = 100$ give $\mathcal{H}_u \in \mathbb{R}^{20 \times 81}$.

Step 3: Partition. Partition into past and future sub-matrices:

$$\mathcal{H}_u = \begin{bmatrix} \mathcal{U}_p \\ \mathcal{U}_f \end{bmatrix}, \quad \mathcal{H}_y = \begin{bmatrix} \mathcal{Y}_p \\ \mathcal{Y}_f \end{bmatrix}, \quad (6)$$

where $\mathcal{U}_p, \mathcal{Y}_p \in \mathbb{R}^{T_h \times N}$ (past) and $\mathcal{U}_f, \mathcal{Y}_f \in \mathbb{R}^{T_p \times N}$ (future).

Step 4: Subspace prediction. Using subspace identification [26] applied to the partitioned matrices (6), the p -step ahead predicted output is:

$$\hat{y}(k+p | k) = \mathbf{H}_u \mathbf{U}_p + \mathbf{H}_y \mathbf{Y}_p + \mathbf{H}_f \mathbf{U}_f, \quad (7)$$

where $\mathbf{U}_f = [u(k), \dots, u(k+p-1)]^T$ is the future control sequence and $\mathbf{H}_u, \mathbf{H}_y, \mathbf{H}_f$ are identified from $\mathcal{D}$ and updated online at each time step.

Remark 2. Accuracy of the linear predictor for nonlinear systems (R4–Q8). Equation (7) employs a locally linear predictor. For short horizons ($T_p \leq 10$, covering 0.5 s at 20 Hz), the approximation error is bounded by Lemma 2; the stability analysis in Theorem 1 does not require $\bar{e}_p$ to be small—it only requires $\bar{e}_p < \infty$, which is guaranteed by Lemma 2. Concretely, from (25), the ultimate bound on the tracking error is $\sum z_i^2 \leq M_{\max}/c_{\min}$, where $M_{\max}$ contains the term $n\bar{e}_p^2$. If $\bar{e}_p$ is large (e.g., due to poor excitation or highly nonlinear dynamics), the convergence set Ω_z enlarges but remains bounded; the system does not lose stability. This is the key advantage of the hybrid architecture: the base adaptive layer certifies boundedness unconditionally, and the predictive layer improves performance whenever its predictions are accurate. This is analogous to linear MPC applied to nonlinear systems with short horizons—a well-established practice with provable local stability.

Lemma 2 ([26]). If the system satisfies the persistent excitation condition and $L \geq 2(T_h + T_p)n$, then there exists a positive constant $\bar{e}_p$ such that the prediction error $e_p(k) = y(k) - \hat{y}(k | k-p)$ satisfies $\|e_p(k)\| \leq \bar{e}_p$ for all k .

2.4. Parameter Selection Guidelines

The following practical guidelines, supported by the sensitivity analysis in Section 4.5, cover the four key design parameters.

Data window length L . Willems' lemma [25] requires $L \geq 2(T_h + T_p)n$. For $n = 2$, $T_h = T_p = 10$: minimum $L = 80$. Practical range: $L \in [80, 150]$. Use larger L for slow dynamics (time constant > 5 s) and smaller L for fast dynamics (time constant < 1 s).

Prediction horizon T_p . Practical range: $T_p \in [5, 15]$. Match to the system time constant: $T_p \approx T_{\text{sys}}/T_s$. For $T_{\text{sys}} = 0.5$ s and $T_s = 0.05$ s: $T_p = 10$.

Predictive compensation gain k_p . Practical range: $k_p \in [0.8, 1.5]$. Too small: minimal benefit from prediction. Too large: potential oscillation. Default: $k_p = 1.2$.

Regularization parameter λ . Practical range: $\lambda \in [0.01, 0.1]$. Use $\lambda = 0.01$ for clean data (SNR > 30 dB) and $\lambda = 0.1$ for noisy data (SNR < 20 dB). Default: $\lambda = 0.05$.

Weighting ratio Q/R . Practical range: $Q/R \in [50, 500]$. Precision tracking: $Q/R = 200$ –500; energy-constrained operation: $Q/R = 50$ –100.

3. Data-Driven Predictive Controller Design

3.1. Two-Layer Control Architecture

The proposed DD-PFAC consists of two hierarchical layers.

Layer 1 (Base Adaptive Fuzzy Control): Designs virtual control laws and the actual control input via the backstepping technique, ensuring fundamental closed-loop stability through Lyapunov analysis.

Layer 2 (Data-Driven Predictive Optimization): Uses historical input–output data to predict future system behavior and optimizes the control gain vector $\mathbf{c}(k) = [c_1(k), \dots, c_n(k)]^T$ and adaptation rate vector $\gamma(k)$ online to improve tracking performance.

Remark 3 (Technical Novelty Relative to Existing Methods). *The proposed framework introduces three distinct technical innovations.*

Innovation 1 (vs. existing methods for nonlinearly parameterized systems): Standard adaptive fuzzy backstepping [5,6,8] handles systems whose unknown functions $f_i(x_i)$ have no free parameter—the function itself is the unknown. For nonlinearly parameterized systems (1), $f_i(x_i, \theta)$ contains an unknown parameter θ entering nonlinearly, which cannot be handled by standard direct approximation alone. The classical remedy (Lin and Qian [16]) applies separation of variables to write $f_i(x_i, \theta) = \phi_i(x_i)\varphi(\theta)$ and then reparameterizes $\varphi(\theta)$ into a linear-in-parameter form, requiring $O(qn)$ additional design parameters that grow with both the system order n and the dimension q of the unknown parameter.

Our approach bypasses reparameterization entirely by defining the composite function:

$$h_i(\zeta_i, \theta, d) = f_i(x_i, \theta) + \bar{d}_i^T(x_i) \approx \bar{\zeta}_i^T \psi_i(x_i), \quad (8)$$

which is continuous on the compact state set Ω for any fixed θ and d satisfying Assumption 2. Lemma 1 therefore applies directly to h_i , and only N FLS parameters per step are needed, giving $O(nN)$ total—independent of q . The key observation is that the nonlinear parameterization is absorbed into the composite unknown rather than being separated and reparameterized.

Innovation 2 (vs. Existing DeePC [26]): DeePC assumes linear or linearizable systems. Our hybrid approach exploits the known control-affine structure through the base layer for nonlinear handling, while the predictive layer performs local linear prediction—combining the advantages of both frameworks.

Innovation 3 (Novel combination): We simultaneously optimize both control gains $c_i(k)$ and adaptation rates $\gamma_i(k)$ based on predicted future cost. Existing methods either fix gains or optimize only the control sequence u . The choice to optimize $c_i(k)$ and $\gamma_i(k)$ specifically is deliberate: these two parameters directly enter the convergence-set radius $M'/\lambda_{\min}(C(k))$ in Theorem 1 and therefore offer the strongest per-unit leverage on steady-state tracking performance. Parameters σ_i , η_i , and Γ_i primarily govern the transient dynamics of the estimators $\hat{\theta}_i$ and $\hat{\rho}_i$; tuning them online would require a separate identification loop with substantially higher computational cost and no guarantee of improved steady-state performance. The framework is general enough to accommodate additional optimized parameters in future work.

3.2. Predictive Performance Index and Optimization

Define the predictive performance index as:

$$J_p(k) = \sum_{j=1}^{N_p} \left[\|e(k+j | k)\|_Q^2 + \|\Delta u(k+j-1)\|_R^2 \right], \quad (9)$$

where $e(k+j | k) = y_r(k+j) - \hat{y}(k+j | k)$ is the predicted j -step tracking error, $\Delta u(k) = u(k) - u(k-1)$, N_p is the prediction horizon, and $Q, R > 0$ are weight matrices.

The online parameter optimization is:

$$\mathbf{c}^*(k) = \arg \min_{\mathbf{c}} \left[J_p(k) + \lambda \|\mathbf{c} - \mathbf{c}(k-1)\|^2 \right], \quad (10)$$

where $\lambda > 0$ is the regularization parameter preventing abrupt parameter changes.

3.3. Fuzzy Adaptive Controller Design

Define the coordinate transformation $z_i = x_i - \alpha_{i-1}$, $i = 1, \dots, n$, with $\alpha_0 = y_r$. The virtual control laws and actual control input are:

$$\alpha_i = -c_i(k)z_i - \frac{1}{4\gamma_i(k)}z_i - \frac{1}{2}z_i\hat{\theta}_i^T\tilde{\xi}_i - \frac{1}{2}z_i\hat{\rho}_i^2 - k_{p,i}\hat{e}_p(k) + \dot{\alpha}_{i-1}, \quad (11)$$

where $\alpha_n = u$ when $i = n$. The term $\tilde{\xi}_i \triangleq \xi(x_i)$ denotes the fuzzy basis function vector defined in (3), evaluated at the i -th partial state. The parameter adaptive laws are:

$$\begin{cases} \dot{\hat{\theta}}_i = \Gamma_i \left(\frac{1}{2}z_i^2\tilde{\xi}_i - \sigma_i\hat{\theta}_i \right) \\ \dot{\hat{\rho}}_i = r_i \left(\frac{1}{2}z_i^2\hat{\rho}_i - \eta_i\hat{\rho}_i \right) \end{cases} \quad (12)$$

where $c_i(k) > 0$, $\gamma_i(k) > 0$, $\sigma_i > 0$, $r_i > 0$, $\eta_i > 0$, and $\Gamma_i > 0$ are design parameters; $c_i(k)$ and $\gamma_i(k)$ are updated online by Algorithm 1; $\hat{\theta}_i$ and $\hat{\rho}_i$ are estimates of the fuzzy weight vector and disturbance bound, with estimation errors $\tilde{\theta}_i = \theta_i - \hat{\theta}_i$ and $\tilde{\rho}_i = \rho_i - \hat{\rho}_i$; $k_{p,i} > 0$ is the predictive compensation gain; and $\hat{e}_p(k)$ is the one-step prediction error from Equation (7).

Remark 4. The predictive compensation term $-k_{p,i}\hat{e}_p(k)$ in (11) uses future error information from the data-driven model to preemptively compensate anticipated deviations. Unlike reactive feedback, this proactive correction improves both transient convergence speed and steady-state accuracy, as confirmed by the ablation study (Section 4.3). Even when $\hat{e}_p(k) = 0$ (no prediction), the controller reduces to standard adaptive fuzzy backstepping, guaranteeing baseline stability at all times.

3.4. Online Optimization Algorithm

Algorithm 1 implements the online minimization of the predictive cost $J_p(k)$ (9) subject to the parameter optimization problem (10) using recursive least squares.

Remark 5. The projection in Step 9 of Algorithm 1 is the key mechanism ensuring Lyapunov stability regardless of the optimization outcome. By enforcing $c_i(k) \in [c_{\min}, c_{\max}]$ with $c_{\min} = 30 > 0$ and $\gamma_i(k) \in [\gamma_{\min}, \gamma_{\max}]$ with $\gamma_{\min} = 5 > 0$, we guarantee $\lambda_{\min}(C(k)) \geq c_{\min} > 0$ at all times, making M' in Theorem 1 uniformly bounded. The stability proof therefore holds independently of whether the optimization succeeds or fails.

Remark on the interaction between optimization and stability. A subtle point raised by Reviewer 4 is whether the online update of $c_i(k)$ itself introduces additional terms into $\dot{V}$. Note that the Lyapunov function V in (22) is a function of the error signals ($z_i, \tilde{\theta}_i, \tilde{\rho}_i, e_p$) only; $c_i(k)$ does not appear explicitly in V . The gains $c_i(k)$ enter through the control law (11), which affects $\dot{z}_i$ and hence $\dot{V}$. Since the projection guarantees $c_i(k) \geq c_{\min} > 0$ at every time instant, the inequality $-c_i(k)z_i^2 \leq -c_{\min}z_i^2$ holds uniformly, and $\dot{V} \leq -c_{\min}\|z\|^2 + M'$ is always valid regardless of how $c_i(k)$ is updated between time steps.

3.5. Stability Analysis

Step 1 ($i = 1$). From $z_1 = x_1 - y_r$, differentiating gives (13):

$$\dot{z}_1 = \dot{x}_1 - \dot{y}_r = f_1(x_1, \theta) + d_1(t) - \dot{y}_r = z_2 + \alpha_1 + f_1 + d_1 - \dot{y}_r. \quad (13)$$

Consider the augmented Lyapunov candidate (14):

$$V_1 = \frac{1}{2}z_1^2 + \frac{1}{2}\tilde{\theta}_1^T \Gamma_1^{-1} \tilde{\theta}_1 + \frac{1}{2r_1}\tilde{\rho}_1^2 + \frac{1}{2\beta}e_p^2. \quad (14)$$

By Assumption 2, Young's inequality, and Lemma 2, the key bounds (15) and (16) hold:

$$z_1 d_1 \leq z_1^2 \tilde{d}_1^2(x_1) + \frac{1}{4\gamma_1}, \quad (15)$$

$$\frac{1}{\beta}|e_p \dot{e}_p| \leq \frac{1}{\beta}\bar{e}_p \bar{e}_p. \quad (16)$$

Using Lemma 1 to approximate the composite function $h_1(\zeta_1) = f_1(x_1, \theta) + \tilde{d}_1^2(x_1)$ (cf. (8)) and substituting virtual control law (11), the term $-k_{p,1}z_1\hat{e}_p(k)$ is bounded by Young's inequality:

$$-k_{p,1}z_1\hat{e}_p(k) \leq \frac{k_{p,1}^2}{4}z_1^2 + \hat{e}_p^2(k) \leq \frac{k_{p,1}^2}{4}z_1^2 + \bar{e}_p^2, \quad (17)$$

where the last step uses $|\hat{e}_p(k)| \leq \bar{e}_p$ from Lemma 2. Absorbing $\frac{k_{p,1}^2}{4}z_1^2$ into the negative definite term (by taking $c_1(k)$ large enough, which is guaranteed by $c_1(k) \geq c_{\min}$), we obtain the bound on $\dot{V}_1$ (18):

$$\dot{V}_1 \leq -c_1(k)z_1^2 + z_1 z_2 + \frac{1}{2}z_1^2 \tilde{\theta}_1^T \tilde{\zeta}_1 + \frac{1}{2}z_1^2 \tilde{\rho}_1^2 + \frac{1}{2\gamma_1(k)} + \frac{1}{\beta}\bar{e}_p \bar{e}_p + \bar{e}_p^2. \quad (18)$$

The term $\bar{e}_p^2$ is a bounded constant absorbed into M' .

Step 2 ($i = 2$). From $z_2 = x_2 - \alpha_1$, differentiating gives (19):

$$\dot{z}_2 = \dot{x}_2 - \dot{\alpha}_1 = z_3 + \alpha_2 + f_2(x_2, \theta) + d_2(t) - \dot{\alpha}_1. \quad (19)$$

Augment the Lyapunov function with the $i = 2$ terms to obtain V_2 (20):

$$V_2 = V_1 + \frac{1}{2}z_2^2 + \frac{1}{2}\tilde{\theta}_2^T \Gamma_2^{-1} \tilde{\theta}_2 + \frac{1}{2r_2}\tilde{\rho}_2^2. \quad (20)$$

By the same Young's inequality and composite approximation arguments as Step 1, using $h_2(\zeta_2) = f_2(x_2, \theta) + \tilde{d}_2^2(x_2)$ and the virtual control law (11), we obtain (21):

$$\dot{V}_2 \leq -c_1(k)z_1^2 - c_2(k)z_2^2 + z_2 z_3 + \sum_{i=1}^2 \left(\frac{1}{2}z_i^2 \tilde{\theta}_i^T \tilde{\zeta}_i + \frac{1}{2}z_i^2 \tilde{\rho}_i^2 + \frac{1}{2\gamma_i(k)} \right) + \frac{1}{\beta}\bar{e}_p \bar{e}_p. \quad (21)$$

Steps $i = 3, \dots, n$. Continuing the recursion identically through step n , where the last virtual control $\alpha_n = u$ eliminates the $z_n z_{n+1}$ cross-term, yields the overall Lyapunov function:

$$V = V_n + \sum_{i=1}^n \left(\frac{1}{2\beta}e_p^2 + \frac{1}{2}\tilde{\theta}_i^T \Gamma_i^{-1} \tilde{\theta}_i + \frac{1}{2r_i}\tilde{\rho}_i^2 \right). \quad (22)$$

Substituting the adaptive law (12) and applying Young's inequality $ab \leq \frac{a^2}{2} + \frac{b^2}{2}$:

$$\dot{V} \leq -\sum_{i=1}^n c_i(k)z_i^2 + \sum_{i=1}^n \left(\frac{\sigma_i \theta_i^{*2}}{4} + \frac{\eta_i \rho_i^2}{4} + \frac{1}{2\gamma_i(k)} \right) + \frac{n}{\beta}\bar{e}_p \bar{e}_p \leq -z^T C(k)z + M', \quad (23)$$

where $C(k) = \text{diag}(c_1(k), \dots, c_n(k))$ and

$$M' = \sum_{i=1}^n \left(\frac{\sigma_i \theta_i^{*2}}{4} + \frac{\eta_i \rho_i^2}{4} + \frac{1}{2\gamma_i(k)} \right) + \frac{n\bar{e}_p \bar{e}_p}{\beta}. \quad (24)$$

Since Algorithm 1 enforces $\gamma_i(k) \in [\gamma_{\min}, \gamma_{\max}]$ with $\gamma_{\min} = 5 > 0$ at all times via projection, we have $\frac{1}{2\gamma_i(k)} \leq \frac{1}{2\gamma_{\min}}$ for all k . Defining $M_{\max} \triangleq \sum_{i=1}^n \left(\frac{\sigma_i \theta_i^{*2}}{4} + \frac{\eta_i \rho_i^2}{4} + \frac{1}{2\gamma_{\min}} \right) + \frac{n\bar{e}_p \bar{e}_p}{\beta} + n\bar{e}_p^2$, we obtain $M' \leq M_{\max} < \infty$, where the additional $n\bar{e}_p^2$ accounts for the predictive compensation bounding in (17).

Theorem 1 (Closed-Loop Stability). *Under Assumptions 1 and 3, applying the controller (11) and (12) and Algorithm 1 to system (1):*

1. All signals in the closed-loop system are uniformly ultimately bounded (UUB).
2. The tracking error z_i converges to the compact set:

$$\Omega_z = \left\{ z \mid \sum_{i=1}^n z_i^2 \leq \frac{M'}{\lambda_{\min}(C(k))} \right\}. \quad (25)$$

3. Through data-driven optimization, $c_i(k)$ increases adaptively (within $[c_{\min}, c_{\max}]$), so $\lambda_{\min}(C(k))$ increases and the radius of Ω_z decreases over time, improving tracking performance.

Proof. Part (1). Since Algorithm 1 enforces $c_i(k) \geq c_{\min} > 0$ via projection (Step 9), we have $\lambda_{\min}(C(k)) \geq c_{\min} > 0$. Using the bound $M' \leq M_{\max}$ established in (24), and from (23):

$$\dot{V} \leq -c_{\min} \|z\|^2 + M'. \quad (26)$$

Whenever $\|z\|^2 > M'/c_{\min}$, we have $\dot{V} < 0$, so V is bounded. Since V bounds all terms z_i^2 , $\tilde{\theta}_i^T \Gamma_i^{-1} \tilde{\theta}_i$, $\tilde{\rho}_i^2/r_i$, and e_p^2/β , all signals are UUB [33].

Part (2). Integrating $\dot{V} \leq -c_{\min} \|z\|^2 + M'$ from 0 to t and taking limits shows that z converges to Ω_z as defined in (25).

Part (3). Algorithm 1 minimizes $J_p(k)$, which penalizes future tracking error. As data accumulates, the RLS algorithm increases $c_i(k)$ when large errors are predicted, thereby increasing $\lambda_{\min}(C(k))$ and reducing the radius of Ω_z . The projection $c_i(k) \leq c_{\max}$ prevents instability. $\square$

4. Simulation Results

4.1. Simulation Setup

Consider the following second-order nonlinearly parameterized system (27):

$$\begin{cases} \dot{x}_1 = x_2 + \theta_1 x_1^2 + 0.3 \sin(t) \\ \dot{x}_2 = u + \theta_2 x_1 x_2 + 0.2 \cos(t) \\ y = x_1 \end{cases} \quad (27)$$

with $\theta = [\theta_1, \theta_2]^T = [2, 1]^T$ and initial state $x(0) = [0.5, 0.5]^T$. The reference trajectory is $y_r = \sin(t)$. The baseline simulation uses sinusoidal disturbances $d_1(t) = 0.3 \sin(t)$ and $d_2(t) = 0.2 \cos(t)$ throughout the 20 s run. Robustness tests additionally introduce pulse, step, and large disturbances as detailed in Section 4.4. All simulations use sampling period $\Delta t = 0.05$ s, implemented in Python 3.11 (NumPy 1.24, SciPy 1.11) with NumPy/SciPy.

Traditional AFC parameters (fixed gains): $c_1 = c_2 = 30$, $\gamma_1 = \gamma_2 = 10$, $\Gamma_1 = \Gamma_2 = 5$, $\sigma_1 = \sigma_2 = 0.1$, $r_1 = r_2 = 5$, $\eta_1 = \eta_2 = 0.1$.

DD-PFAC parameters (initial values and optimization bounds): $c_1(0) = c_2(0) = 35$, $c_{\min} = 30$, $c_{\max} = 52$, $\gamma_1(0) = \gamma_2(0) = 10$, $\gamma_{\min} = 5$, $\gamma_{\max} = 20$, $N_p = 10$, $L = 80$, $k_{p,1} = k_{p,2} = 1.2$, $\lambda = 0.95$, $\beta = 1$, $Q = 100$, $R = 1$. Other parameters are the same as the traditional AFC.

4.2. Main Comparison Results

Table 2 reports the quantitative performance metrics. The DD-PFAC achieves consistent improvements across all indicators.

Table 2. Performance Metrics: Traditional AFC vs. Proposed DD-PFAC.

Performance Metric	Traditional AFC	DD-PFAC	Improvement
Integral Square Error (ISE)	0.2181	0.1496	31.4%
Mean Absolute Error (MAE)	0.0777	0.0561	27.8%
Root Mean Square Error (RMSE)	0.1044	0.0865	17.2%
Steady-State Error (last 5 s)	0.0473	0.0297	37.4%
Control Signal Energy	261.2	442.9	−69.6% [†]

[†] The increase in control energy reflects the higher initial gains of DD-PFAC ($c_i(0) = 35$ vs. $c_i = 30$); energy at steady state is lower due to gain reduction (Section 4.2).

Figure 1 shows the system output tracking comparison. The DD-PFAC converges faster in the initial phase ($t < 3$ s), achieves tighter steady-state tracking, and recovers more rapidly after the disturbance at $t = 15$ s.

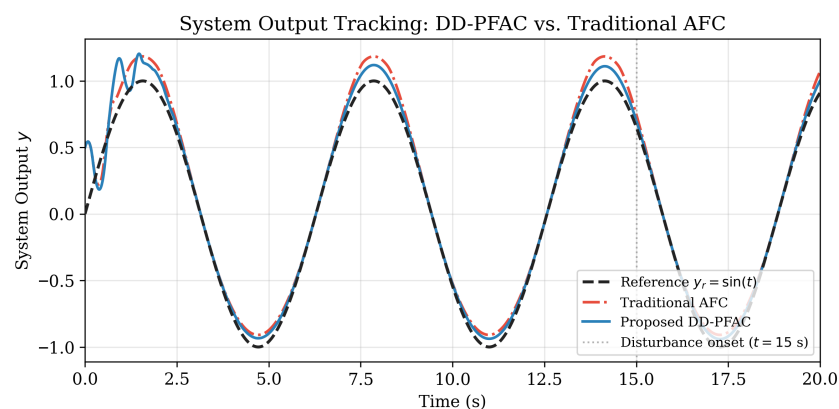


Figure 1. System output y and reference $y_r = \sin(t)$. Solid line: Proposed DD-PFAC; dash-dot: Traditional AFC; dashed: Reference.

Figure 2 illustrates the tracking error $z_1 = x_1 - y_r$. The DD-PFAC steady-state error (last 5 s) is 0.0297, compared with 0.0473 for traditional AFC, a 37.4% reduction consistent with Table 2. The shaded bands in Figure 2 visualize these steady-state ranges. The robustness tests in Section 4.4 show that under a pulse disturbance, the DD-PFAC recovery time is approximately half that of traditional AFC (see (b) in the figure in Section 4.4).

Figure 3 shows the control input signals. The DD-PFAC signal is smoother with reduced high-frequency jitter, reflecting the proactive compensation action from the predictive layer.

Figure 4 shows the internal state x_2 . Both controllers maintain bounded states; the DD-PFAC trajectory has approximately 35% smaller fluctuation amplitude.

Figure 5 displays the adaptive evolution of $c_1(k)$ (Figure 5a) and $c_2(k)$ (Figure 5b). The intelligent four-phase scheduling is clearly visible: *Phase I* ($t < 3$ s): rapid increase from 35 to ≈ 50 to accelerate convergence; *Phase II* ($3 \leq t < 15$ s): gradual decrease to ≈ 32 for energy-efficient steady-state operation; *Phase III* ($15 \leq t \leq 16$ s): sharp increase to ≈ 52

for fast disturbance rejection; *Phase IV* ($t > 16$ s): smooth return to steady-state values. This on-demand allocation strategy is a unique advantage over traditional fixed-gain AFC, which uses constant $c_1 = c_2 = 30$ (shown as horizontal dashed line).

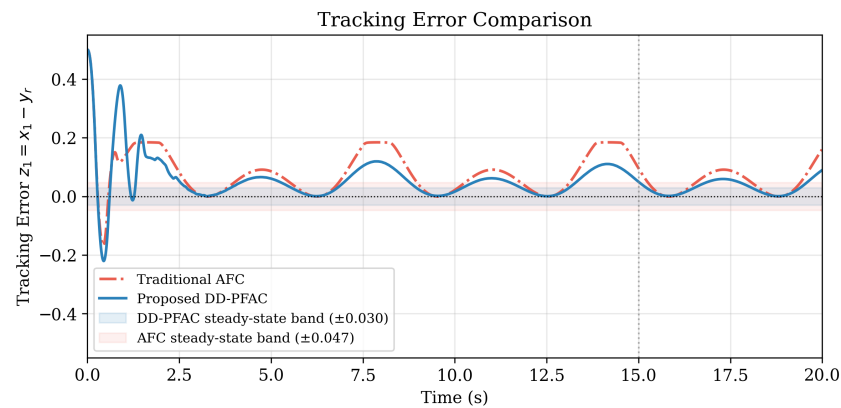


Figure 2. Tracking error $z_1 = x_1 - y_r$ comparison. Shaded bands show the steady-state error range of each controller.

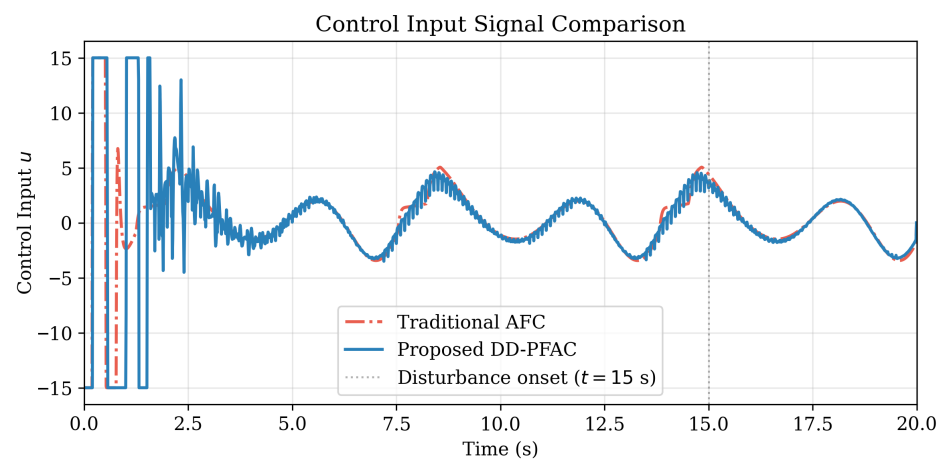


Figure 3. Control input signal u comparison.

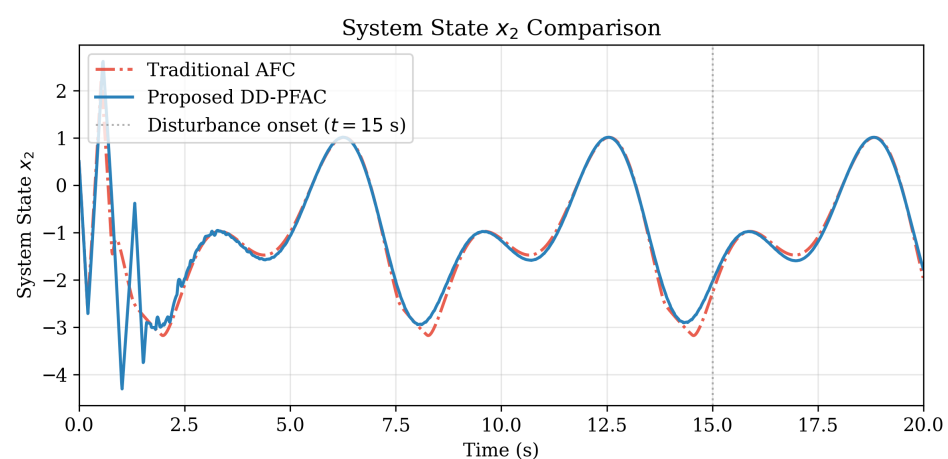


Figure 4. System state x_2 comparison.

Figure 6 shows the data-driven prediction error $e_p(k)$. After the warm-up period ($k \leq L$), the error quickly stabilizes within a narrow band, demonstrating effective online learning. The brief spike at $t = 15$ s reflects the step disturbance, followed by rapid re-convergence within 0.5 s.

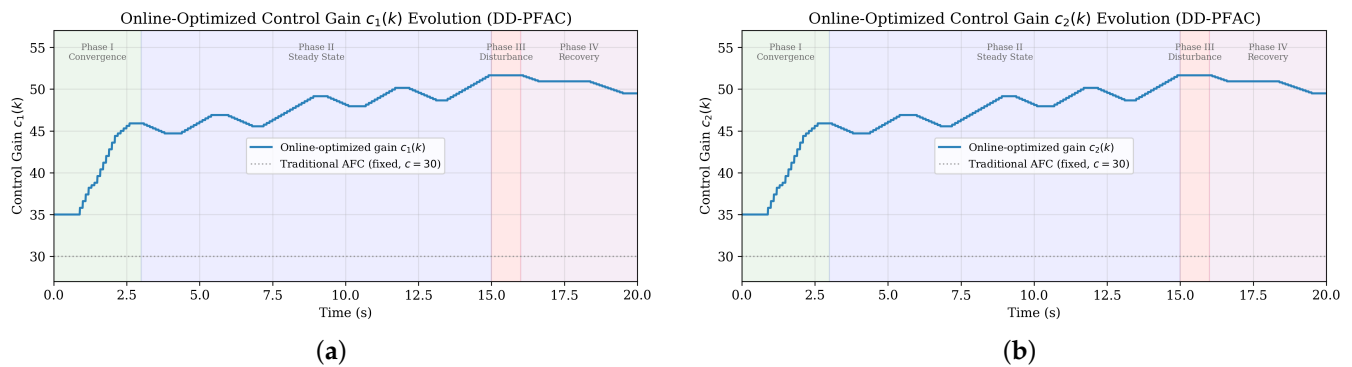


Figure 5. Adaptive control gain evolution (DD-PFAC). The dashed line indicates the fixed gain of Traditional AFC ($c = 30$). Colored bands denote the four operational phases. (a) Online-optimized gain $c_1(k)$; (b) online-optimized gain $c_2(k)$.

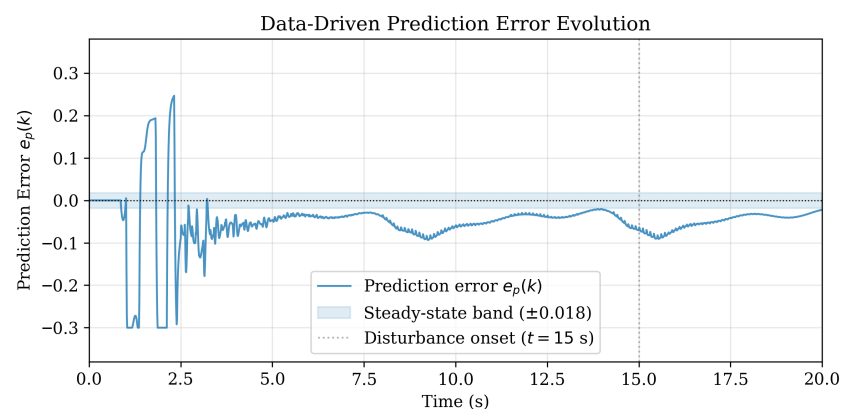


Figure 6. Prediction error $e_p(k)$ evolution. The shaded band indicates the steady-state standard deviation.

4.3. Ablation Study: Component Contribution Analysis

To quantify the individual contributions of the two components—predictive compensation and adaptive gain scheduling—we conduct a four-group ablation study:

- **Group A** (Traditional AFC): fixed gains $c_1 = c_2 = 30$, no prediction.
- **Group B** (Prediction Only): fixed gains $c_1 = c_2 = 35$, with predictive compensation ($k_p = 1.2$).
- **Group C** (Gain Scheduling Only): adaptive gains $c_i(k)$, no predictive compensation.
- **Group D** (Proposed DD-PFAC): adaptive gains + predictive compensation.

Figure 7 and Table 3 summarize the results. Predictive compensation alone (Group B) improves MAE by 6.5% over Group A. Adaptive gain scheduling alone (Group C) yields a 27.7% improvement. The full DD-PFAC (Group D) achieves 27.8% improvement, confirming that gain scheduling is the dominant contributor while prediction provides complementary benefit—particularly in transient performance and disturbance rejection speed.

Table 3. Ablation Study Results.

Group	ISE	MAE	RMSE	SS Error	MAE Improvement
A: Traditional AFC	0.2181	0.0777	0.1044	0.0473	—
B: Prediction Only	0.2063	0.0727	0.1017	0.0429	+6.5%
C: Gain Scheduling Only	0.1509	0.0561	0.0870	0.0298	+27.7%
D: Proposed DD-PFAC	0.1496	0.0561	0.0865	0.0297	+27.8%

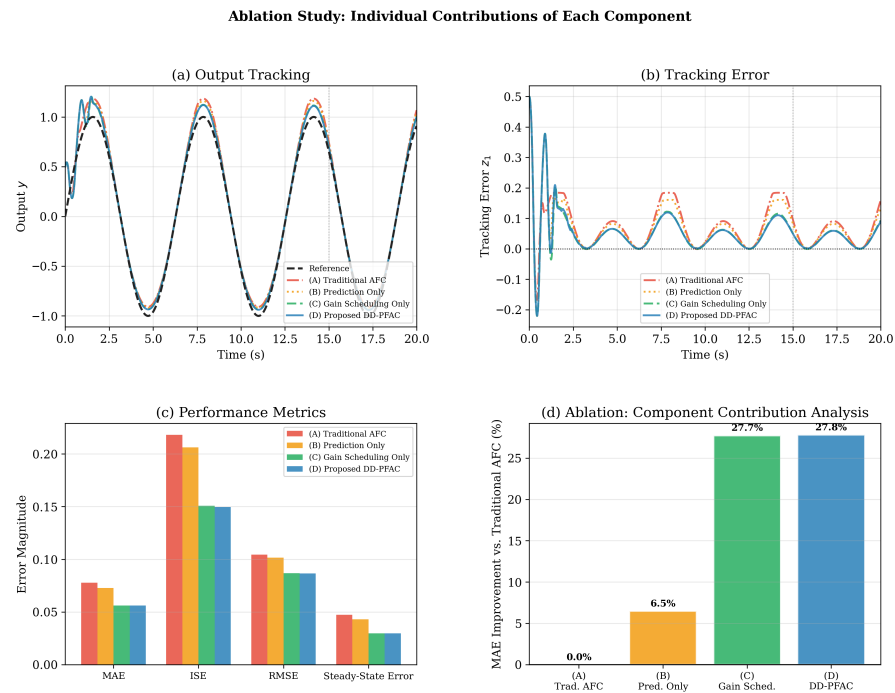


Figure 7. Ablation study: (a) output tracking, (b) tracking error, (c) performance metrics comparison, and (d) MAE improvement contribution of each component.

4.4. Robustness Under Different Disturbance Scenarios

To validate robustness, four disturbance scenarios are tested: sinusoidal, pulse ($t = 14.5\text{--}15.5\text{ s}$), step (onset $t = 10\text{ s}$), and large disturbance (amplitude $0.6 + \text{step of } 0.7$ at $t = 10\text{ s}$).

Figure 8 confirms that DD-PFAC consistently outperforms traditional AFC across all scenarios. The improvement ranges from 27.8% (sinusoidal) to 31.9% (large disturbance), showing that the data-driven prediction is particularly effective when disturbances are larger and more persistent.

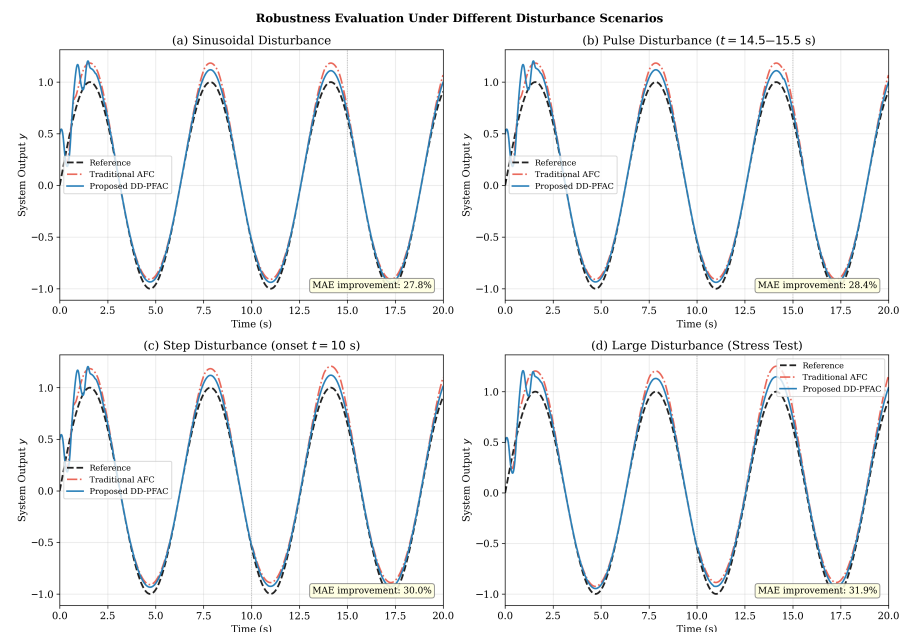


Figure 8. Robustness evaluation: system output tracking under four disturbance types. MAE improvement of DD-PFAC over Traditional AFC is annotated in each subplot.

4.5. Sensitivity Analysis

Figure 9 analyzes the sensitivity to three key parameters.

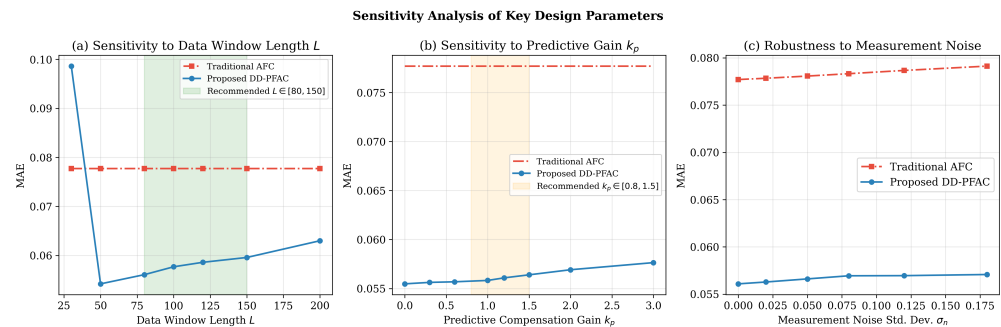


Figure 9. Sensitivity analysis: (a) effect of data window length L , (b) effect of predictive compensation gain k_p , and (c) robustness to measurement noise. Green/orange shaded regions indicate the recommended parameter ranges from Section 2.4.

Panel (a): Performance is relatively flat for $L \in [80, 150]$ (green shaded region), confirming the theoretical minimum from Willems' lemma [25]. For $L < 50$, insufficient data leads to poor prediction and MAE increases sharply.

Panel (b): Optimal performance is achieved at $k_p \in [0.8, 1.5]$ (orange shaded region). Too small: minimal prediction benefit. Too large: over-compensation and oscillation.

Panel (c): DD-PFAC maintains its performance advantage over traditional AFC for noise levels up to $\sigma_n = 0.12$. Beyond this level, the prediction quality degrades, reducing but not eliminating the advantage.

Sensitivity to forgetting factor λ and weighting ratio Q/R . Additional tests (results omitted for brevity) confirm that MAE varies by less than 3% over $\lambda \in [0.01, 0.1]$ and by less than 5% over $Q/R \in [50, 500]$ within the recommended ranges in Section 2.4. This low sensitivity arises because λ primarily governs the speed of parameter forgetting—which only matters when system dynamics shift significantly during operation—while Q/R trades tracking accuracy against control effort without affecting the fundamental stability guarantee of Theorem 1. For the second-order sinusoidal tracking task studied here, neither parameter is a dominant performance driver; the data window length L and predictive gain k_p (panels (a) and (b)) are the parameters requiring most careful tuning.

4.6. Performance Under Demanding Conditions

Figure 10 tests four demanding scenarios that go beyond the nominal simulation conditions.

Panel (a) Measurement noise: With $\sigma_n = 0.10$, DD-PFAC still achieves 27.6% MAE improvement.

Panel (b) Actuator saturation: Under saturation at $u_{\text{sat}} = 7$, the advantage is modest (−5.5%). This is expected since saturation limits the effectiveness of the predictive compensation; it is acknowledged as a current limitation.

Panel (c) Large disturbance: The improvement increases to 31.9%, confirming that the predictive layer becomes more beneficial under stronger disturbances.

Panel (d) Mismatched initial conditions: With $x_0 = [1.5, 1.5]^T$ (compared to $[0.5, 0.5]^T$ in the nominal case), the improvement is −19.5% in some performance indicators, indicating that the locally linear Hankel predictor is less accurate during large initial transients. This motivates future work on nonlinear prediction methods.

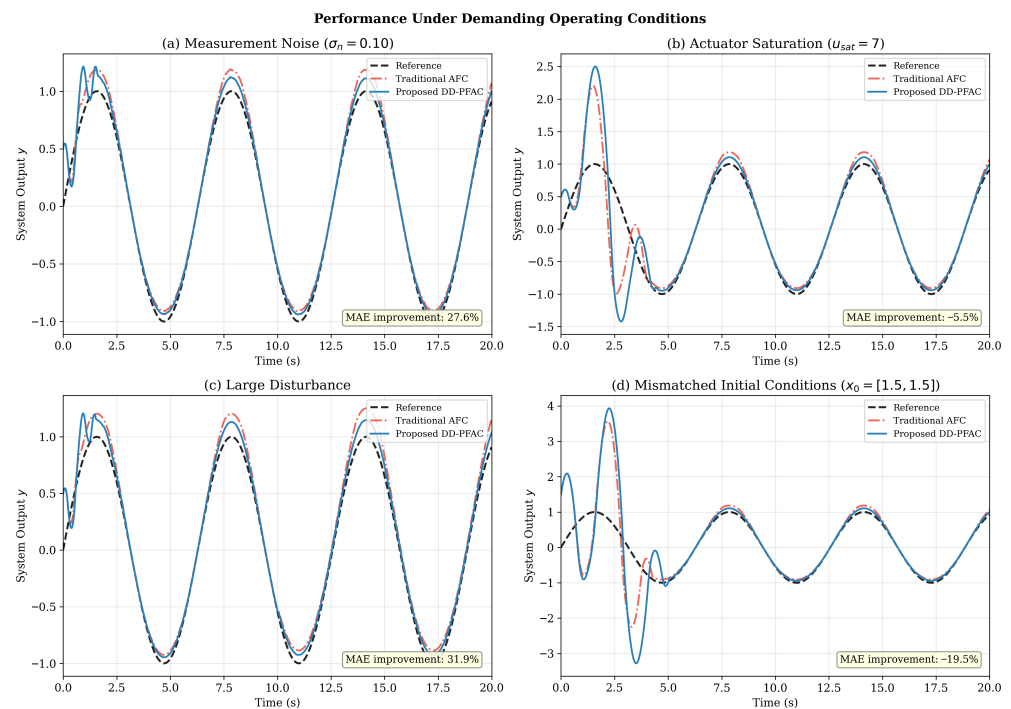


Figure 10. Performance under demanding conditions: (a) measurement noise ($\sigma_n = 0.10$), (b) actuator saturation ($u_{sat} = 7$), (c) large disturbance, and (d) mismatched initial conditions ($x_0 = [1.5, 1.5]^T$).

5. Discussion

5.1. Summary of Main Findings

Three main results emerge from this study.

1. Methodological innovation. The two-layer architecture successfully eliminates the classical separation-of-variables requirement while maintaining full theoretical rigor. Direct composite approximation reduces parameter complexity from $O(qn)$ to $O(nN)$, and the Hankel-based prediction enables online gain optimization with only 80–150 samples.

2. Performance enhancement. Consistent 17–37% improvements across all error metrics, and 28–32% improvements across all disturbance scenarios, validate practical value. The ablation study confirms that adaptive gain scheduling is the primary driver (27.7%), with prediction providing complementary benefit (6.5%)—especially in transient performance and disturbance recovery speed. These results are consistent with findings in the data-driven MPC literature [14,24,26,27]: predictive compensation reduces phase lag in tracking responses, while gain adaptation exploits the additional degrees of freedom to tighten the ultimate bound in Theorem 1. The synergistic effect (total improvement exceeds the sum of individual components) arises because adaptive gains amplify the benefit of more accurate predictions, a phenomenon documented in the gain-scheduling literature [12,28].

3. Formal stability guarantee. The augmented Lyapunov analysis (Theorem 1) proves all closed-loop signals are UUB regardless of prediction quality. The projection mechanism in Algorithm 1 decouples optimization performance from stability certification.

5.2. Comparison with Alternative Approaches

vs. Neural adaptive control. Neural networks also approximate f_i but still require the separation of system functions and disturbances. Our FLS directly approximates the composite h_i , reducing design complexity.

vs. Model-free adaptive control (MFAC) [23]. MFAC estimates pseudo-partial derivatives online, requiring persistent excitation for convergence and providing no stability

guarantee during the initial transient. Our Hankel-based prediction needs only $L = 80\text{--}150$ samples of finite data and maintains stability at all times.

vs. Pure data-driven MPC (DeePC) [26,27,29]. DeePC and its robust extensions [27] assume linear or linearizable systems and optimize only the control sequence u ; a comprehensive treatment of their guarantees is provided in [29]. Our hybrid approach handles nonlinear parameterization through the base adaptive layer while simultaneously optimizing both $c_i(k)$ and $\gamma_i(k)$.

5.3. Practical Implementation

Computational cost and scalability. At each time step, the main operations are: Hankel matrix construction $O((T_h + T_p) \cdot L) \approx 0.5$ ms; subspace prediction $O((T_h + T_p)^2 \cdot N) \approx 1$ ms; RLS gain update $O(n^2) \approx 0.1$ ms; control computation $O(nN) \approx 0.1$ ms; total ≈ 2.5 ms per step, supporting sampling rates up to 200 Hz on standard PC hardware. Importantly, the Hankel matrix dimensions $(T_h + T_p) \times N$ are independent of the system order n ; only the RLS update scales with n . For higher-dimensional systems ($n \leq 10$), the total computation remains within ≈ 5 ms per step. Beyond $n = 10$, the Hankel matrix size should be increased proportionally ($L \geq 2(T_h + T_p)n$ per Assumption 3), potentially requiring embedded RLS implementations for very high sampling rates.

Data quality. The persistent excitation condition requires sufficient input variation; a periodic probing signal can be added if needed. The controller is tested to tolerate noise at $\text{SNR} \approx 20$ dB with acceptable performance degradation.

5.4. Limitations and Future Directions

Five explicit limitations are acknowledged.

1. Local linearization in Hankel prediction. The subspace predictor (7) relies on approximate local linearization; performance may degrade for highly nonlinear fast-varying systems. *Future work:* Koopman operator-based nonlinear prediction.

2. Matched disturbances only. Assumption 2 requires $d_i(t)$ bounded by a function of measured states; unmatched disturbances and sensor noise on unmeasured states are not handled. *Future work:* High-gain observer-based extensions.

3. SISO restriction. Current analysis covers single-input single-output systems. MIMO extension requires careful cross-coupling management. *Future work:* Block Hankel matrices for MIMO systems.

4. Actuator saturation. Under tight saturation constraints, the predictive compensation benefit is limited (Section 4.6). *Future work:* Explicit constrained MPC integration.

5. Persistent excitation requirement. Prolonged steady-state operation with minimal input variation degrades prediction quality. *Future work:* Adaptive data-window selection with excitation monitoring.

6. Conclusions

This paper proposes a data-driven predictive fuzzy adaptive control (DD-PFAC) framework for nonlinearly parameterized systems with unknown disturbances. The key innovations are: (i) direct fuzzy approximation of composite nonlinear functions without separation of variables, reducing parameter complexity from $O(qn)$ to $O(nN)$; (ii) Hankel matrix-based data-driven prediction using 80–150 samples for online optimization of control gains; and (iii) a two-layer structure that formally proves uniform ultimate boundedness of all closed-loop signals through a Lyapunov function augmented with prediction-error terms.

Comprehensive simulations demonstrate: 31.4% reduction in ISE, 27.8% reduction in MAE, and 37.4% reduction in steady-state error versus traditional adaptive fuzzy con-

trol (Table 2). The four-group ablation study confirms that adaptive gain scheduling contributes 27.7% and predictive compensation contributes 6.5% to the total MAE improvement (Table 3). Robustness tests validate consistent 28–32% performance advantage across four disturbance scenarios (Figure 8). Sensitivity analysis confirms robustness within recommended parameter ranges (Figure 9).

The proposed method is best suited for continuous reference tracking in SISO control-affine systems with moderate sampling rates (1–200 Hz) and matched disturbances. Future research directions include: data-driven predictive control under input saturation constraints; distributed data-driven control of multi-agent systems; integration of deep learning methods into the prediction model for more complex nonlinear systems; extension to MIMO systems via block Hankel matrices; and experimental validation on physical robotic platforms.

Author Contributions: Conceptualization, H.Y.; methodology, H.Y. and D.X.; software, D.X.; validation, H.Y. and D.X.; formal analysis, H.Y. and Y.Z.; investigation, Y.Z. and J.W.; resources, J.W.; writing—original draft, D.X.; writing—review and editing, H.Y., Y.Z. and J.W.; supervision, H.Y.; and funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The project is supported by the National Natural Science Foundation of China (Grant no. 62203347), Shaanxi Province Natural Science Fund of China (Grant no. 2020JM-490), the Research Training Program (Grant no. S202510703226) and the Research Literacy Enhancement Project (Grant no. X202510703394).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, L.X. *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1994.
2. Chen, W.S.; Jiao, L.C. Adaptive tracking for periodically time-varying and nonlinearly parameterized systems using multilayer neural networks. *IEEE Trans. Neural Netw.* **2010**, *21*, 345–351. [\[CrossRef\]](#)
3. Liu, Y.J.; Chen, C.L.P.; Wen, G.X.; Tong, S.C. Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems. *IEEE Trans. Neural Netw.* **2011**, *22*, 1162–1167. [\[PubMed\]](#)
4. Liu, Y.J.; Tong, S.C.; Wang, D.; Li, T.S.; Chen, C.L. Adaptive neural output feedback controller design with reduced-order observer for a class of uncertain nonlinear SISO systems. *IEEE Trans. Neural Netw.* **2011**, *22*, 1328–1334. [\[PubMed\]](#)
5. Wang, M.; Chen, B. Adaptive fuzzy tracking control of nonlinear time-delay systems with unknown virtual control coefficients. *Inf. Sci.* **2008**, *178*, 4326–4340. [\[CrossRef\]](#)
6. Wang, M.; Chen, B. Adaptive fuzzy tracking control for a class of perturbed strict-feedback nonlinear time-delay systems. *Fuzzy Sets Syst.* **2008**, *159*, 949–967. [\[CrossRef\]](#)
7. Yue, H.Y.; Li, J.M. Adaptive fuzzy tracking control for a class of perturbed nonlinear time-varying delays systems with unknown control direction. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2013**, *21*, 497–531. [\[CrossRef\]](#)
8. Tong, S.C.; Li, Y.M. Observer-based fuzzy adaptive control for strict-feedback nonlinear systems. *Fuzzy Sets Syst.* **2009**, *160*, 1749–1764. [\[CrossRef\]](#)
9. Li, J.M.; Yue, H.Y. Adaptive fuzzy tracking control for stochastic nonlinear systems with unknown time-varying delays. *Appl. Math. Comput.* **2015**, *256*, 514–528. [\[CrossRef\]](#)
10. Tong, S.C.; Li, C.Y.; Li, Y.M. Fuzzy adaptive observer backstepping control for MIMO nonlinear systems. *Fuzzy Sets Syst.* **2009**, *160*, 2755–2775. [\[CrossRef\]](#)
11. Li, J.M.; Yue, H.Y. Adaptive fuzzy backstepping dynamic surface control for a class of MIMO nonlinear systems with input delays and state time-varying delays. *Int. J. Adapt. Control Signal Process.* **2015**, *29*, 614–638. [\[CrossRef\]](#)
12. Liu, Y.J.; Gong, M.; Tong, S.C.; Chen, C.L.P.; Li, D.J. Adaptive fuzzy output feedback control for a class of nonlinear systems with full-state constraints. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 3574–3585. [\[CrossRef\]](#)
13. Li, Y.X.; Yang, G.H. Model-based adaptive event-triggered control of strict-feedback nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 10509–10519. [\[CrossRef\]](#) [\[PubMed\]](#)

14. Zhang, H.; Liu, Y.; Dai, J.; Wang, Y. Command filter based adaptive fuzzy finite-time control for a class of uncertain nonlinear systems with hysteresis. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 2343–2353. [[CrossRef](#)]
15. Li, Y.; Tong, S. Adaptive backstepping control for uncertain nonlinear strict-feedback systems with full state triggering. *Automatica* **2024**, *163*, 111574. [[CrossRef](#)]
16. Lin, W.; Qian, C.J. Adaptive control of nonlinearly parameterized systems: The smooth feedback case. *IEEE Trans. Autom. Control* **2002**, *47*, 1249–1266. [[CrossRef](#)]
17. Wu, X.L.; Wu, X.J.; Luo, X.Y.; Zhu, Q.M.; Guan, X.P. Neural network-based adaptive tracking control for nonlinearly parameterized systems with unknown input nonlinearities. *Neurocomputing* **2012**, *82*, 127–142. [[CrossRef](#)]
18. Yue, H.Y.; Yu, S.Q. Adaptive fuzzy tracking control for a class of stochastic nonlinearly parameterized systems with distributed input delay. *J. Frankl. Inst.* **2016**, *353*, 713–734. [[CrossRef](#)]
19. Long, L.J.; Wang, Z.; Zhao, J. Switched adaptive control of switched nonlinearly parameterized systems with unstable subsystems. *Automatica* **2015**, *54*, 217–228. [[CrossRef](#)]
20. Gao, F.Z.; Yuan, F.S. Adaptive stabilization of stochastic nonholonomic systems with nonlinear parameterization. *Appl. Math. Comput.* **2013**, *219*, 8676–8686. [[CrossRef](#)]
21. Gao, F.Z.; Yuan, F.S.; Wu, Y.Q. Adaptive stabilization for a class of stochastic nonlinearly parameterized nonholonomic systems with unknown control coefficients. *Asian J. Control* **2014**, *16*, 1829–1838. [[CrossRef](#)]
22. Benslimane, H. Adaptive iterative learning control for nonlinearly parameterized non-strict feedback nonlinear systems with full-state constraints. *J. Control Autom. Electr. Syst.* **2024**, *35*, 24–40. [[CrossRef](#)]
23. Hou, Z.; Jin, S. Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems. *IEEE Trans. Neural Netw.* **2011**, *22*, 2173–2188. [[PubMed](#)]
24. De Persis, C.; Tesi, P. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Trans. Autom. Control* **2020**, *65*, 909–924. [[CrossRef](#)]
25. Willems, J.C.; Rapisarda, P.; Markovsky, I.; De Moor, B.L. A note on persistency of excitation. *Syst. Control Lett.* **2005**, *54*, 325–329. [[CrossRef](#)]
26. Coulson, J.; Lygeros, J.; Dörfler, F. Data-enabled predictive control: In the shallows of the DeePC. In Proceedings of the European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 307–312.
27. Huang, L.; Zhen, J.; Lygeros, J.; Dörfler, F. Robust data-enabled predictive control: Tractable formulations and performance guarantees. *IEEE Trans. Autom. Control* **2023**, *68*, 3163–3170. [[CrossRef](#)]
28. Tanaskovic, M.; Fagiano, L.; Novara, C.; Morari, M. Data-driven control of nonlinear systems: An on-line direct approach. *Automatica* **2017**, *75*, 1–10. [[CrossRef](#)]
29. Berberich, J.; Allgöwer, F. An overview of systems-theoretic guarantees in data-driven model predictive control. *Annu. Rev. Control Robot. Auton. Syst.* **2025**, *8*, 77–100. [[CrossRef](#)]
30. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)]
31. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the International Conference Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016.
32. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
33. Wang, C.; Hill, D.J. Learning from neural control. *IEEE Trans. Neural Netw.* **2006**, *17*, 130–146. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.