drones

MDPI

Article

High-Angle-of-Attack Longitudinal Control of UAVs Using Physics-Informed Neural Networks and RBF-Compensated Nonlinear Dynamic Inversion

Dong Zhang [1,2], Binjie Han [3], Yeguang Wang [2], Shaoze Huo [3], Qingfeng Zhao [3,*] and Kai Liu [3]

[1] Department of Aeronautics and Astronautics, College of Intelligent Robotics and Advanced Manufacturing Innovation, Fudan University, Shanghai 200433, China; 20110290020@fudan.edu.cn
[2] Shengyang Aircraft Design and Research Institute, Shengyang 110035, China; 21110290023@m.fudan.edu.cn
[3] School of Mechanics and Aerospace Engineering, Dalian University of Technology, Dalian 116081, China; hanbinjie@mail.dlut.edu.cn (B.H.); 12556089@mail.dlut.edu.cn (S.H.); carsonliu@dlut.edu.cn (K.L.)
* Correspondence: zhaoqingfeng@dlut.edu.cn

Highlights

What are the main findings?

- The PINN is able to capture variations in the pitching moment coefficient, static stability coefficient, and elevator effectiveness, with MSEs of 0.0001, 0.0110, and 0.0085, respectively.
- The PINN-informed NDI controller achieved approximately 40% pitch-up suppression compared with PID control, while RBF compensation increased this rate to approximately 50% and reduced transient oscillations without noticeably affecting velocity, altitude, or throttle.

What are the implications of the main findings?

- Combining physics-informed identification with online residual compensation provides an effective way to reduce model mismatch and reject disturbances under nonlinear and rapidly changing flight conditions.
- The proposed framework has the potential to support stable attitude regulation during unexpected high-angle-of-attack excursions in civil UAVs, subject to further hardware-in-the-loop and controlled flight validation.

Abstract

This study evaluates a longitudinal control method integrating a physics-informed neural network (PINN), nonlinear dynamic inversion (NDI), and radial basis function (RBF) compensation using an open-source F-16 dynamics model as a numerical benchmark. The pitch dynamics are embedded in the offline training loss as a physical constraint. During control operation, the trained PINN is evaluated at the current inputs with fixed weights and biases, and its predictions update the inverse model and control effectiveness terms. The RBF network provides online residual compensation. The PINN captures the trends of the pitching moment coefficient and its aerodynamic derivatives, providing model information for dynamic inversion. Compared with the implemented fixed-gain PID baseline, the model-informed controllers reduce angle-of-attack tracking errors and transient overshoot under the prescribed simulation conditions. RBF compensation further reduces the remaining tracking deviations. These results concern the prescribed numerical tracking test; they do not establish flight safety, loss-of-control prevention, emergency stall recovery, or performance on a civil aircraft. The stability result is conditional on a bounded

Check for updates

Academic Editor: Pablo Rodríguez-Gonzálvez

Received: 23 September 2026
Revised: 3 November 2026
Accepted: 3 November 2026
Published: 12 November 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) license.

operating region, a uniformly bounded residual, effective control authority, and the stated actuator realization conditions.

Keywords: UAV; high angle of attack; physics-informed neural network; nonlinear dynamic inversion; radial basis function neural network; numerical validation

1. Introduction

High-angle-of-attack longitudinal control is a nonlinear tracking problem with potential relevance to flight stability research. Research on high-angle-of-attack perching trajectories also illustrates the relevance of these dynamics to controlled landing [1]. Here, the demonstrated contribution is restricted to longitudinal tracking in a numerical benchmark. Flow separation, leading-edge vortex evolution, vortex breakdown, and distorted inflow over control surfaces can make the aerodynamic forces and moments nonlinear and unsteady, and can be associated with stall and pitch-up behavior [2,3]. Though these considerations motivate the control problem, they are not evidence that the present simulations establish loss-of-control prevention.

The difficulty of high-angle-of-attack flight control arises not only from complex nonlinear aerodynamic characteristics but also from the amplification of model errors and external disturbances under extreme flight conditions. Gusts, aerodynamic parameter perturbations, control effectiveness degradation, sensor noise, actuator dynamics, and unmodeled unsteady aerodynamic forces may all cause deviations between the ideal control input computed by the controller and the actual UAV response. Previous studies have shown that high-angle-of-attack unsteady aerodynamic modeling [4], high-angle-of-attack experiments [5], and data-driven stability and control analysis [6,7] are important approaches for investigating complex flight states. However, relying solely on offline aerodynamic databases or piecewise linear models may be insufficient for reliable attitude regulation during unexpected high-angle-of-attack excursions.

A fixed-gain PID controller provides a simple and reproducible tracking baseline, but a single set of gains need not preserve the same performance when aerodynamic characteristics change. This observation should not be generalized to all PID-based methods. Hybrid PID designs can combine PID feedback with nonlinear model-based control, as in the PID–SDRE quadrotor architecture studied by Giernacki et al. [8]. Fuzzy gain scheduling adjusts PID gains using a rule-based mechanism [9]; adaptive PID and neural network-based PID have also been investigated for quadrotor control [10,11]. These methods address operating point dependence, uncertainty, or nonlinear behavior in ways that are absent from a conventional fixed-gain baseline. Their application settings differ from the high-angle-of-attack longitudinal benchmark considered here, and the present study does not establish superiority over those methods.

NDI instead uses an inverse dynamics model to prescribe error dynamics when the relevant control map is sufficiently accurate and remains invertible [12]. Its performance is sensitive to model mismatch, small control effectiveness, and actuator constraints; external disturbance rejection is not guaranteed by inversion alone. The rationale for the present combination is to update the inverse model using PINN identification and compensate for remaining uncertainty using an RBF network. Any performance advantage reported below is relative to the implemented PID baseline under the specified numerical conditions and common evaluation metrics.

For UAVs with multiple control surfaces, the pseudoinverse of the control effectiveness matrix can be used to allocate the desired generalized control moments to different control surfaces, avoiding the difficulty of directly inverting a non-square control effectiveness

matrix [13]. Tang investigated high-angle-of-attack thrust vector control using an improved incremental nonlinear dynamic inversion method, resulting in improved attitude control capability of aircraft under high-angle-of-attack conditions [14]. Tian et al. designed a dynamic inversion-based flight control method for high-angle-of-attack supersonic civil aircraft and analyzed its command tracking performance [15]. Zhu et al. investigated the coordinated allocation of aerodynamic direct force and thrust vector control inputs using nonlinear dynamic inversion [16]. Stougie et al. combined incremental nonlinear dynamic inversion with flight envelope protection for the safe control of a flying-V configuration aircraft [17]. To address variations in the control effectiveness of novel control surfaces, Stam and de Visser further proposed an adaptive incremental nonlinear control allocation method [18].

However, the control performance of pseudoinverse-based NDI depends strongly on the accuracy of the aircraft model and control effectiveness matrix. When aerodynamic parameters such as lift, drag, pitching moment, and control surface effectiveness used by the controller contain errors, the model inversion result deviates from the actual system, resulting in dynamic inversion errors. This may reduce tracking accuracy, lead to unreasonable control surface allocation, induce control oscillations, or even cause closed-loop instability [19]. In addition, under high-angle-of-attack conditions, aerodynamic parameters vary rapidly with angle of attack and flight state, and the control effectiveness matrix may become ill-conditioned or nearly singular, further reducing the reliability of the pseudoinverse solution. Therefore, improving the accuracy of aerodynamic parameter identification is essential for enhancing the control performance of pseudoinverse-based NDI.

Aerodynamic parameter identification is an important approach for obtaining high-fidelity flight dynamics models. Traditional identification methods mainly include the equation error method, output error method, filter error method, least squares method, maximum likelihood estimation, and Kalman filter-based methods. These methods are theoretically mature, provide parameters with clear physical meanings, and offer strong engineering interpretability. As a result, they have been widely used in flight test data modeling and aircraft parameter estimation [20]. Among them, the least squares method and its recursive form have high computational efficiency and are suitable for online applications, while the extended Kalman filter can handle nonlinear problems to a certain extent and frequency-domain identification methods are advantageous in noise resistance, model validation, and flight quality analysis. Morelli et al. conducted systematic studies on real-time frequency-domain identification and flight data modeling [21]. In recent years, methods such as flight test-based longitudinal mode and aerodynamic parameter identification, aerodynamic modeling and parameter identification for high-speed vehicles, and regularized regression identification have further promoted the application of identification techniques to complex aircraft. To meet the parameter estimation requirements of different aircraft, Hu et al. studied aerodynamic modeling and aerodynamic parameter identification for high-speed vehicles [22]; Lerro et al. identified aircraft longitudinal modes and aerodynamic parameters using flight test data [23]; and Kumar and Ghosh applied regularized regression to aerodynamic parameter estimation from flight data to reduce the influence of parameter correlation and data noise on identification results [24]. Although traditional identification methods are theoretically mature, physically interpretable, and engineering-oriented, they usually require a predefined aerodynamic model structure and impose relatively strict requirements on data excitation, noise levels, and model form. In the presence of strong nonlinearity, unsteadiness, and abrupt aerodynamic changes at high angles of attack, such fixed-structure low-order parameter models cannot always maintain high identification accuracy.

With the development of artificial intelligence, neural networks and intelligent optimization methods have begun to be applied to high-angle-of-attack aerodynamic modeling. Cao et al. proposed an interpretable machine learning identification method for unknown aircraft dynamics, which can account for the influence of control inputs and enable online identification [25]. For high-angle-of-attack multi-degree-of-freedom coupled maneuvers, Dong et al. developed a feature-level attention recurrent neural network based on wind tunnel test data, improving the accuracy, interpretability, and generalization capability of unsteady aerodynamic force prediction [26]. Ma et al. combined a residual transformer autoencoder with physical mechanisms for aircraft high-angle-of-attack dynamics modeling [27]. However, such methods still depend on the quantity and distribution of training samples as well as on their coverage of the flight state space. When samples are insufficient or actual operating conditions exceed the training range, increased extrapolation errors and inadequate physical consistency may occur. Therefore, introducing flight dynamics constraints into neural network modeling is an important approach for improving the reliability of aerodynamic parameter identification.

Physics-informed neural networks (PINNs) introduce flight dynamics equations, aerodynamic constraints, and physical residuals into the neural network loss function, thereby combining data fitting capability with physical laws. Compared with traditional identification methods, PINNs do not rely entirely on predefined linear or low-order parameter models and can better describe the complex nonlinear variations in aerodynamic parameters at high angles of attack. Compared with purely data-driven neural networks, PINNs can maintain better physical consistency under limited samples or measurement noise, improving the credibility and generalization capability of the identification results [28–30]. In recent years, physics-informed data-driven methods have been applied to nonlinear unsteady aerodynamic modeling at high angles of attack, demonstrating their potential for complex aerodynamic identification. Liu et al. applied a physics-informed data-driven method to nonlinear unsteady aerodynamic modeling at high angles of attack and improved the modeling capability for complex aerodynamic responses by integrating observation data with aerodynamic mechanisms [31]. In the implementation considered here, the PINN is trained offline and evaluated at the current flight state during control operation. Its predictions update the aerodynamic model and control effectiveness terms without online optimization of the PINN weights or biases.

Although PINNs can improve aerodynamic model accuracy, residual errors between the identified model and the actual UAV remain unavoidable because of limited training data, sensor noise, actuator dynamics, and unmodeled high-order aerodynamic effects. In addition, external disturbances such as gusts cannot be fully eliminated through aerodynamic parameter identification alone. Therefore, an additional compensation mechanism is required on the basis of PINN and pseudoinverse-based NDI. To handle model uncertainty and external disturbances, existing studies commonly adopt robust control, adaptive control, sliding mode control, extended state observers, disturbance observers, and neural network compensation methods [32,33]. Studies on event-triggered learning control, extended state observer-based generalized predictive control, and neural network-based dynamic flight envelope protection have shown that during post-stall pitch maneuvers and high-angle-of-attack flight, the control system must possess both nonlinear processing capability and disturbance rejection capability [34,35].

Disturbance rejection beyond constant offsets is also important because external disturbances encountered during flight can vary with time. Robust interconnection and damping assignment passivity-based control (IDA-PBC) has been investigated for non-separable port-controlled Hamiltonian (PCH) systems subject to time-varying external disturbances [36]. Robust cascaded control with anti-saturation has also been studied for fixed-wing trajectory

tracking [37], highlighting the importance of actuator constraints in robust flight control design. These studies motivate an explicit description of the disturbances represented by the compensation model and the conditions and limitations of the tracking error guarantee.

Neural networks have strong nonlinear approximation capability and can estimate compound disturbances and unmodeled dynamics that are difficult to describe using accurate mathematical models. Among different types of neural networks, radial basis function (RBF) neural networks have the advantages of simple structure, strong local approximation capability, easily adjustable output weights, and relatively low computational cost, making them suitable for real-time UAV control loops [38]. Compared with fixed-structure observers that rely on predefined disturbance models, RBF compensation does not require accurate prior knowledge of the disturbance form. Compared with structurally complex deep neural networks, its parameter update is more lightweight. Therefore, an RBF neural network can be used to estimate and compensate online for PINN identification residuals, unmodeled aerodynamic forces, and external disturbances, further improving the control accuracy and robustness of pseudoinverse-based NDI under complex high-angle-of-attack conditions.

To address the model mismatch considered here, this study combines offline PINN training with online model evaluation, model-based NDI, and online RBF residual compensation. The identified parameters update the inverse model, while the RBF term compensates for the remaining uncertainty in the modeled pitch channel. The numerical comparison evaluates the combined method and its uncompensated counterpart against the implemented PID baseline. Performance and robustness statements are restricted to the tested conditions and the assumptions specified in the analysis.

The contribution evaluated here is a model-informed longitudinal controller with online residual compensation. Its evidence consists of parameter identification results and tracking responses in the specified F-16 numerical benchmark. Safety-related applications remain motivations for future investigation rather than demonstrated outcomes.

2. Longitudinal Dynamics Modeling of the UAV

The open-source F-16 dynamics model is used solely as a numerical benchmark for longitudinal flight stability research; Figure 1 identifies the reference configuration.



Figure 1. Reference F-16 configuration associated with the numerical dynamics benchmark.

During flight, deflections of the control surfaces and changes in attitude cause aerodynamic forces to act on the UAV. In the longitudinal plane, these forces mainly include drag F_{xq} and lift F_{yq}. The aerodynamic moment generated in the body-fixed coordinate system is the pitching moment M_{zq}, and these aerodynamic forces and moments vary with the flight state of the UAV. The drag, lift, and pitching moment of the UAV are expressed as follows [39]:

$$\begin{cases} L = C_L(V, h, \alpha, \delta_e)\, \bar{q} S, \\ D = C_D(V, h, \alpha, \beta, \delta_e)\, \bar{q} S, \\ M = C_m(V, h, \alpha, \delta_e)\, \bar{q} S l, \end{cases} \tag{1}$$

where C_L, C_D, and C_m denote the dimensionless longitudinal aerodynamic force and moment coefficients, δ_e is the elevator deflection angle, $\bar{q}$ is the dynamic pressure, S is the wing reference area, l is the mean aerodynamic chord, V is the UAV velocity, h is the flight altitude, and α and β respectively denote the angle of attack and sideslip angle.

The longitudinal equations below are derived from the reference dynamics model for the numerical control study. Because the inertial and body-fixed reference frames describe the UAV motion differently, the following relationship is obtained during coordinate transformation:

$$\frac{d\mathbf{V}}{dt} = \frac{\delta \mathbf{V}}{\delta t} + \boldsymbol{\Omega} \times \mathbf{V} \tag{2}$$

where $\frac{d\mathbf{V}}{dt}$ denotes the absolute derivative of the velocity vector $\mathbf{V}$ in the ground-fixed coordinate system, $\frac{\delta \mathbf{V}}{\delta t}$ denotes the relative derivative of $\mathbf{V}$ in the velocity coordinate system, and Ω denotes the angular velocity vector. Therefore, in the body-fixed coordinate system, the longitudinal centroid translational dynamics can be established as follows:

$$\begin{cases} \dot{V}_z = -V_x q + \dfrac{L - mg\cos\vartheta}{m}, \\ \dot{V}_x = V_z q + \dfrac{D + T - mg\sin\vartheta}{m}, \end{cases} \tag{3}$$

where V_z and V_x are the velocity components of the UAV along the longitudinal axes of the body-fixed frame, ϑ is the pitch angle, m is the UAV mass, and g is the gravitational acceleration. Similarly, the rotational dynamics of the UAV about the center of mass can be established in the body-fixed frame. When only longitudinal motion is considered, it is expressed as

$$\dot{q} = \frac{M}{I_y}, \tag{4}$$

where q is the pitch angular rate of the UAV and I_y is the pitch moment of inertia.

After obtaining the equations for the angular rate and velocity components, the angular and linear displacement equations are described in the ground-fixed and body-fixed coordinate systems, respectively. According to the coordinate transformation relationship and the kinematic equations, the relations can be obtained as described below.

The displacement and altitude kinematic equations in the ground-fixed coordinate system are

$$\begin{cases} \dot{x}_g = V_x \cos\vartheta - V_z \sin\vartheta, \\ \dot{h} = V_x \sin\vartheta + V_z \cos\vartheta, \end{cases} \tag{5}$$

where x_g and h denote the forward displacement and vertical altitude of the UAV in the ground-fixed coordinate system, respectively; thus, the longitudinal dynamic equations of the numerical benchmark can be written as follows:

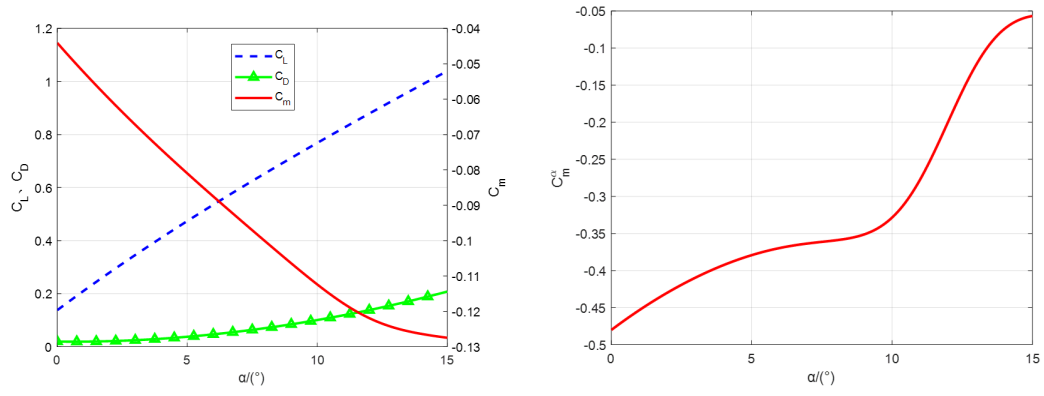
$$\begin{cases} \dot{V}_z = -V_x q + \dfrac{L - mg\cos\vartheta}{m}, \\ \dot{V}_x = V_z q + \dfrac{D + T - mg\sin\vartheta}{m}, \\ \dot{q} = \dfrac{M}{I_y}, \\ \dot{x}_g = V_x\cos\vartheta - V_z\sin\vartheta, \\ \dot{h} = V_x\sin\vartheta + V_z\cos\vartheta, \\ \dot{\vartheta} = q. \end{cases} \tag{6}$$

To simulate aerodynamic uncertainty under high-angle-of-attack conditions, the pitching moment coefficient C_m is perturbed under the high-angle-of-attack condition. This perturbation is a numerical test of parameter identification accuracy and control robustness under aerodynamic model mismatch, which is relevant to maintaining stable attitude regulation; it is not a physical turbulence model. The other lateral-directional characteristics remain consistent with those of the baseline model. The specific perturbation model is given by Equation (7):

$$C_m = \begin{cases} C_{m_0} + C_m^q q + C_z(x_{cgr} - x_{cg}) + 0.2\alpha, & \alpha < 10, \\ C_{m_0} + C_m^q q + C_z(x_{cgr} - x_{cg}) + 0.2\alpha + 0.23\alpha, & \alpha \geq 10, \end{cases} \tag{7}$$

where C_{m_0} is the zero-angle-of-attack pitching moment coefficient, C_m^q is the pitch damping coefficient, q is the pitch angular rate, C_z is the coefficient associated with the center-of-gravity position, x_{cgr} is the reference center-of-gravity position, x_{cg} is the actual center-of-gravity position, and α is the angle of attack.

The variations with respect to the angle of attack of the UAV lift, drag, pitching moment, and corresponding aerodynamic coefficients and static stability coefficient are shown in Figure 2. These data provide the basis for the subsequent controller design and stability evaluation.



(**a**) (**b**)

Figure 2. Aerodynamic characteristic curves under high-angle-of-attack conditions: (**a**) variations in lift, drag, and pitching moment coefficients with angle of attack and (**b**) variation in the static stability coefficient with angle of attack.

As shown in the figure, the lift coefficient C_L increases approximately linearly with the angle of attack, whereas the drag coefficient C_D exhibits a nonlinear increase, indicating that drag growth becomes more pronounced at high angles of attack. The pitching moment

coefficient C_m decreases overall, reflecting a certain nose-down restoring moment. The static stability coefficient remains negative over the entire angle-of-attack range, indicating that the UAV is longitudinally statically stable. However, when the angle of attack exceeds 10°, its absolute value decreases rapidly, suggesting a significant reduction in the longitudinal static stability margin under high-angle-of-attack conditions. Therefore, active control is required to maintain the flight attitude.

3. Control Law Design for Pitch-Up Suppression of UAVs at High Angles of Attack

The controller is formulated for longitudinal regulation using the numerical benchmark described in Section 2. The control objective is to reduce tracking deviations and transient pitch-up in the prescribed test. An offline-trained PINN supplies aerodynamic model information to the NDI controller through online inference with fixed network parameters, while an RBF network compensates for remaining modeled-channel uncertainty. Figure 3 shows the identification, inverse control, compensation, and control allocation pathways. Their numerical evaluation does not constitute validation of a system for flight envelope protection or preventing loss of control.

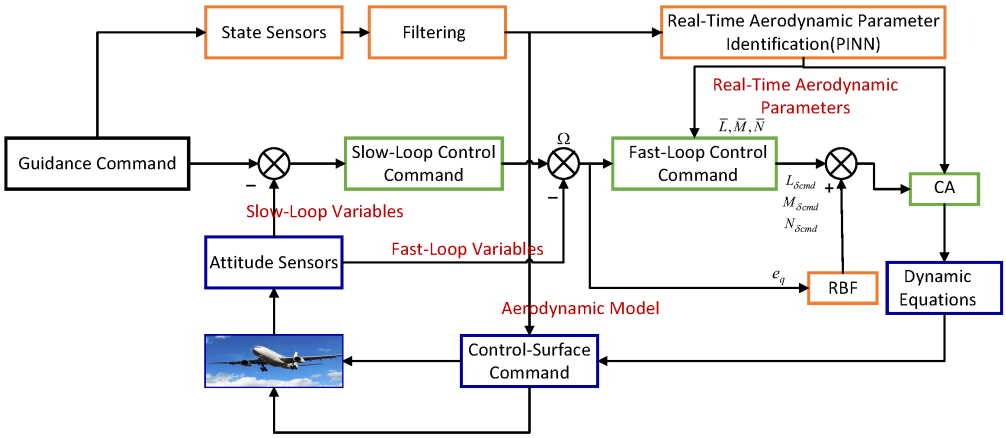


Figure 3. Block diagram of NDI control with online inference from an offline-trained PINN and RBF compensation.

The formulation assumes a supplied reference and effective actuator authority, but does not implement an emergency stall recovery system or demonstrate stability during severe turbulence; these applications would require separate reference generation, envelope, sensing, and actuator assessments.

As shown in Figure 3, the control algorithm adopts a cascaded structure consisting of a slow loop and a fast loop. The slow loop generates the pitch rate command from the attitude tracking error, and the fast loop uses angular rate feedback and the current aerodynamic model to calculate the control moment through NDI. Filtered flight state signals are supplied to the offline-trained PINN for online inference; its weights and biases remain fixed during closed-loop operation. The predicted pitching moment coefficient and aerodynamic derivatives provide model information for fast-loop control and control allocation. The measured pitch rate q is the sole input to the RBF basis functions, whereas the tracking error $e_q = q - q_c$ drives the output-weight adaptation law. The corresponding RBF moment correction is combined with the fast-loop moment command before control allocation. The resulting control surface commands drive the aircraft model, and the measured states close the feedback loop.

3.1. Angle-of-Attack Controller Design Based on Nonlinear Dynamic Inversion

Nonlinear dynamic inversion (NDI) uses a system model to compute an input associated with prescribed state-derivative dynamics. Approximate feedback linearization requires an adequately accurate model and an effective control map. NDI alone does not establish general disturbance rejection superiority over PID-based methods; the identification and residual compensation components are evaluated separately in this study. The inner-loop model for the pitch rate is introduced below.

At high angles of attack, only the longitudinal motion of the unmanned aerial vehicle (UAV) is considered. The angular rate dynamics are expressed as follows:

$$\dot{q} = \frac{1}{I_y} M = \frac{1}{I_y} (M_0 + M_c) \tag{8}$$

where I_y is the pitch moment of inertia, M is the total pitching moment, M_0 denotes the uncontrolled pitching moment, and M_c denotes the control pitching moment.

The pitch rate q is selected as the inner-loop state variable, i.e., $x_1 = q$, and the control moment M_c about the body pitch axis is selected as the inner-loop control input, i.e., $u_1 = M_c$. Accordingly, using the state-space formulation and NDI theory, Equation (8) can be rewritten as

$$\dot{x}_1 = \dot{q} = f_1(x_1) + g_1 u_1, \tag{9}$$

where $f_1(x_1)$ is the drift term of the inner-loop model, representing the component of the state dynamics that does not explicitly depend on the control input u_1. The input gain g_1 represents the effect of a unit control input on the rate of change of the state. In the simplified single-axis model adopted in this study, the control pitching moment is treated as the control input, and effects such as cross-axis coupling are neglected. Therefore,

$$\begin{cases} f_1(x_1) = \dfrac{M_0}{I_y}, \\ g_1 = \dfrac{1}{I_y}, \\ u_1 = M_c. \end{cases} \tag{10}$$

For the inner-loop control law design, the pitch rate command is calculated by the slow loop, and the desired value of the pitch angular acceleration on the left-hand side of Equation (9) is specified as follows:

$$\dot{x}_{1c} = \dot{q}_c = K_q (q_c - q) \tag{11}$$

where K_q is the bandwidth of the pitch rate channel, q_c is the pitch rate command, and $\dot{q}_c$ denotes the prescribed desired pitch angular acceleration. Using dynamic inversion to achieve the desired form specified above, the fast-loop control law is expressed as

$$\begin{aligned} u_1 = M_c &= g_1^{-1} [\dot{q}_c - f_1(x_1)] \\ &= I_y K_q (q_c - q) - M_0. \end{aligned} \tag{12}$$

The slow loop is the attitude-angle loop and surrounds the inner loop. Its input is the output of a higher-level command generator, while its output serves as the input to the inner-loop controller. The governing equation of the slow loop is given by

$$\dot{\vartheta} = q. \tag{13}$$

The pitch angle ϑ is selected as the outer-loop state variable, i.e., $x_2 = \vartheta$, and the pitch rate command q_c is selected as the outer-loop virtual control input, i.e., $u_2 = q_c$. Under the assumption of ideal inner-loop tracking, $q = q_c$, the above equation can also be rewritten as

$$\dot{x}_2 = \dot{\vartheta} = f_2(x_2) + g_2 u_2, \tag{14}$$

where

$$\begin{cases} f_2(x_2) = 0, \\ g_2 = 1, \\ u_2 = q_c. \end{cases} \tag{15}$$

For the outer-loop control law design, the desired value of $\dot{\vartheta}$ on the left-hand side is calculated from the external command ϑ_c as follows:

$$\dot{x}_{2c} = \dot{\vartheta}_c = K_\vartheta(\vartheta_c - \vartheta) \tag{16}$$

where K_ϑ is the bandwidth of the pitch-angle channel and $\dot{\vartheta}_c$ denotes the prescribed desired pitch angle rate. To achieve the desired form specified in Equation (16), the slow-loop dynamic inversion control law is expressed as

$$u_2 = q_c = g_2^{-1}\dot{\vartheta}_c = K_\vartheta(\vartheta_c - \vartheta). \tag{17}$$

Thus, time-scale separation is applied to the pitch control channel. The inner- and outer-loop NDI control laws are designed separately and connected in cascade, yielding the control law expressions for both loops. Substituting Equation (17) into Equation (12) gives the final pitching moment control command:

$$M_{c,\mathrm{cmd}} = I_y K_q[K_\vartheta(\vartheta_c - \vartheta) - q] - M_0 \tag{18}$$

where $M_{c,\mathrm{cmd}}$ is the commanded control pitching moment and the term $-M_0$ compensates for the uncontrolled pitching moment.

Finally, according to the elevator control effectiveness, the elevator deflection command can be calculated as

$$\delta_e = \frac{M_{\delta\mathrm{cmd}}}{\bar{q}SlC_m^{\delta_e}}, \tag{19}$$

where $\bar{q}$ is the dynamic pressure.

The control law derivation shows that the solution of the elevator deflection command δ_e depends on the elevator control effectiveness $C_m^{\delta_e}$. Under high-angle-of-attack conditions, aerodynamic parameters may vary significantly. If fixed or biased aerodynamic parameters are still used, the control surface command computed by dynamic inversion will deviate from the actual control demand, reducing the tracking accuracy and stability of the system. Therefore, to improve the model matching capability of the NDI controller under high-angle-of-attack conditions, it is necessary to accurately identify the pitching moment coefficient and elevator control effectiveness. On this basis, a physics-informed neural network is introduced in this study. By embedding flight dynamics constraints into the data-driven model, the accuracy and physical consistency of aerodynamic parameter identification can be improved.

3.2. PINN-Based Aerodynamic Parameter Identification Method

PINNs are developed from deep neural networks (DNNs) [40]. In this study, the PINN is trained offline using model-generated data. Forward propagation evaluates the network output, while backpropagation and optimization update its parameters during offline training only. After deployment, the weights and biases remain fixed and only forward evaluations are performed. The forward prediction process is expressed as follows:

$$\begin{cases} h_{(1)} = \sigma_1(w_1 X + b_1), \\ h_{(l+1)} = \sigma_l(w_i h_{(l)} + b_i), \quad l = 2, \ldots, n-2, \\ \hat{Y} = \sigma_{n-1}(w_{n-1} h_{(n-1)} + b_{n-1}) \end{cases} \tag{20}$$

where σ_l denotes the activation function of each layer, w_i denotes the weight matrix of each layer, and b_i denotes the bias of each layer.

The fitting error loss function of the neural network is defined as $\mathcal{L}$. During error backpropagation, the loss gradients with respect to the network weights and biases are calculated according to Equation (21):

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w_i} = \dfrac{\partial \mathcal{L}}{\partial \sigma_i}\dfrac{\partial \sigma_i}{\partial w_i} = \dfrac{\partial \mathcal{L}}{\partial \sigma_i}\sigma_i'\sigma_{i-1}, \\ \dfrac{\partial \mathcal{L}}{\partial b_i} = \dfrac{\partial \mathcal{L}}{\partial \sigma_i}\dfrac{\partial \sigma_i}{\partial b_i} = \dfrac{\partial \mathcal{L}}{\partial \sigma_i}\sigma_i', \quad 1 \le i \le n-1, \\ \dfrac{\partial \mathcal{L}}{\partial \sigma_i} = \dfrac{\partial \mathcal{L}}{\partial \sigma_{i+1}}\dfrac{\partial \sigma_{i+1}}{\partial \sigma_i}. \end{cases} \tag{21}$$

By constructing a hierarchical network architecture and integrating multiple hidden layers, the DNN significantly improves its ability to extract abstract data features and exhibits strong nonlinear mapping capability and fitting accuracy.

A PINN combines neural networks with physical equations by introducing physical constraints into the conventional loss function [41]. In this way, the constraints of the physical system are incorporated into the network training process, enabling the network to learn patterns from data while ensuring compliance with physical laws. This improves the subsequent interpretability of the network. The structure of the PINN algorithm is shown in Figure 4.

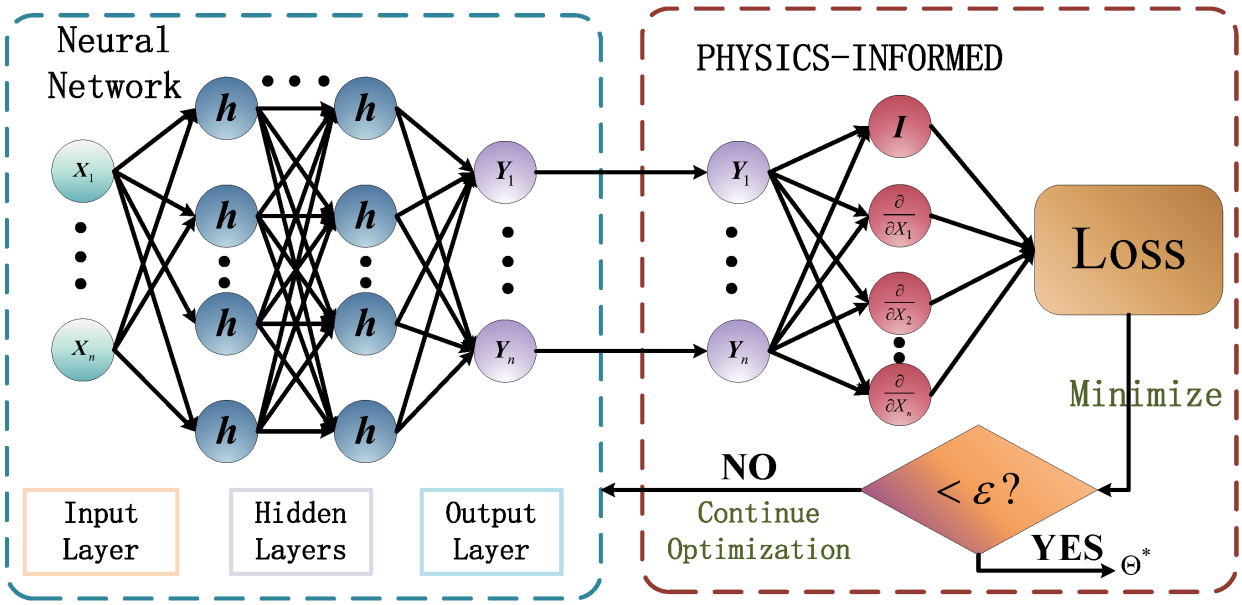


Figure 4. Physics-informed neural network diagram.

The loss function of the PINN consists of two components: the data loss and the physics loss. In this study, the PINN loss is defined using the mean squared error (MSE) metric, as follows:

$$\mathcal{L}_{\mathrm{PINN}}(\Theta) = \lambda_1 \mathrm{MSE}_{\mathrm{da}} + \lambda_2 \mathrm{MSE}_{\mathrm{ph}} \tag{22}$$

where $\mathrm{MSE}_{\mathrm{da}}$ is the data fitting loss and $\mathrm{MSE}_{\mathrm{ph}}$ is the physics constraint loss. They are specifically defined as follows:

$$\begin{cases} \mathrm{MSE}_{\mathrm{ph}} = \dfrac{1}{N_{\mathrm{ph}}} \displaystyle\sum_{i=1}^{N_{\mathrm{ph}}} \left| \mathcal{F}(X_{\mathrm{ph}}^i, Y_{\mathrm{ph}}^i, t_{\mathrm{ph}}^i) \right|^2, \\ \mathrm{MSE}_{\mathrm{da}} = \dfrac{1}{N_{\mathrm{da}}} \displaystyle\sum_{i=1}^{N_{\mathrm{da}}} \left| \hat{Y}(X_{\mathrm{da}}^i) - Y^i \right|^2. \end{cases} \tag{23}$$

The activation function after the output layer is set as an identity function, and the final neural network representation is given by Equation (24):

$$Y(X;\Theta) = (f_n \circ \sigma \circ f_{n-1} \circ \cdots \circ \sigma \circ f_1)(X) \tag{24}$$

where the operator '$\circ$' denotes the composite operator, f_n is the transformation function acting on the vector, $\Theta = \{W_1, W_2, \ldots, b_1, b_2, \ldots\}$ represents the trainable parameters of the network, $Y(X;\Theta)$ is the network output, and X is the input.

Offline PINN training seeks parameters that minimize the loss through the optimization problem in Equation (25):

$$(W_1^*, W_2^*, \ldots, b_1^*, b_2^*, \ldots) = \underset{(W_1,\ldots,b_1,\ldots)}{\arg\min}\ \mathcal{L}_{\mathrm{PINN}}(\Theta; X, Y). \tag{25}$$

During offline training, the optimization problem is approximately solved using an iterative gradient descent framework. The network parameters are updated along the negative loss gradient direction as follows:

$$\Theta^{m+1} = \Theta^m - \eta \nabla_\Theta \mathcal{L}_{\mathrm{PINN}}(\Theta^m; X, Y) \tag{26}$$

where $\eta > 0$ is the learning rate and $\mathcal{L}_{PINN}(\Theta^m; X, Y)$ is the loss function at the m-th iteration.

A physics-informed neural network requires physical information to be introduced into the loss function. In the identification and fitting process of the PINN, the physical information related to the pitching moment coefficient should be specified. The corresponding expression is extracted as follows:

$$\begin{cases} I_y \dot{q} = M, \\ M = C_m \bar{q} S l. \end{cases} \tag{27}$$

By rearranging the above equations, the pitch angular acceleration can be expressed as:

$$\dot{q} = \frac{C_m \bar{q} S l}{I_y} \tag{28}$$

The pitch rate is taken directly from the offline state samples; no numerical time differentiation of the sampled q is required to construct this physics loss. For a current sample X_i, the network-predicted coefficient $\widehat{C}_m(X_i;\Theta)$ is substituted into Equation (28) to obtain the predicted angular acceleration. A forward Euler step with $\Delta t = 0.01$ s then predicts the next pitch rate, and the physics loss compares that prediction with the corresponding next state sample:

$$\begin{aligned} \hat{\dot{q}}_i &= \frac{\widehat{C}_m(X_i;\Theta)\, \bar{q}_i S l}{I_y}, \\ \hat{q}_{i+1|i} &= q_i + \Delta t\, \hat{\dot{q}}_i, \qquad \Delta t = 0.01 \text{ s}, \\ \mathrm{MSE}_{\mathrm{ph}} &= \frac{1}{N_{\mathrm{ph}}} \sum_{i=1}^{N_{\mathrm{ph}}} \left| \hat{q}_{i+1|i} - q_{i+1} \right|^2 \end{aligned} \tag{29}$$

where N_{ph} is the number of admissible adjacent-sample pairs, while q_i and q_{i+1} are the measured or model-generated pitch rate samples from the same trajectory, separated by 0.01 s. The

pair index i enumerates such pairs rather than arbitrary neighboring rows in a shuffled dataset. Dynamic pressure and the known geometric and inertial quantities are used in consistent units. The q samples enter directly, while $\dot{q}$ is calculated from the predicted moment coefficient rather than being estimated by finite differences of q. Δt denotes the sampling interval adopted for the offline physics loss.

The data loss function of the deep-learning network is defined as

$$\mathrm{MSE}_{\mathrm{da}} = \frac{1}{N_{\mathrm{da}}} \sum_{i=1}^{N_{\mathrm{da}}} \left| \hat{C}_m(X_{\mathrm{da}}^i) - C_m^i \right|^2. \tag{30}$$

By combining the data loss and the physics loss, the final PINN loss function is obtained as

$$\mathcal{L}_{\mathrm{PINN}} = \lambda_1 \mathrm{MSE}_{\mathrm{da}} + \lambda_2 \mathrm{MSE}_{\mathrm{ph}}, \tag{31}$$

where λ_1 and λ_2 are the weights of the data loss and physics loss, respectively. The inputs of the PINN are the angle of attack, pitch angular rate, and elevator deflection, and the output is the pitching moment coefficient. Equation (29) specifies the one-step numerical integration of the angular acceleration in Equation (28) used in the physics loss. Reference moment coefficients are supervision targets for the data loss; the physics loss prediction uses the network-predicted coefficient and the state samples.

3.3. Angle-of-Attack Pitch-Up Suppression Controller Design Based on RBF Neural Network Compensation

To reduce tracking errors associated with uncertain pitching moments, the dynamic inversion controller is augmented with online compensation. The objective is to reduce transient tracking deviations in the modeled pitch channel under bounded residual uncertainty. This is a control performance objective, not a demonstrated flight safety benefit.

The offline-trained PINN supplies aerodynamic model information and the RBF network provides residual compensation. Pitch rate q is selected as the sole physical input to the RBF network because it is directly available as feedback in the compensated channel. This single-input choice reduces the basis function dimension and online computation, but represents only the disturbance component associated with pitch rate under the considered conditions. Other state-dependent or explicitly time-varying components remain in the residual $\varepsilon(x,t)$ in Equation (34). The network output estimates a disturbance in the equation for the pitch angular acceleration; its equivalent compensating pitching moment is $M_{\mathrm{RBF}} = -I_y \hat{f}_{\mathrm{un}}$ under the sign convention in Equation (35). The basis function input q is distinct from $e_q = q - q_c$, which drives the output-weight adaptation law in Equation (43). Figures 3 and 5 show these signals and the compensation pathway.

The RBF neural network consists of three layers: an input layer, a hidden layer, and an output layer. The input layer receives the input feature vector and transmits it to the hidden layer. The hidden layer processes the input through radial basis functions defined by specific centers and variances, and generates the hidden-layer output. In each hidden-layer neuron, the Euclidean distance between the input vector and the corresponding center vector is first calculated, then the Gaussian radial basis function is evaluated using this distance to generate the neuron output. Finally, the output layer linearly combines the outputs of the hidden layer and processes them through the activation function to obtain the final output.

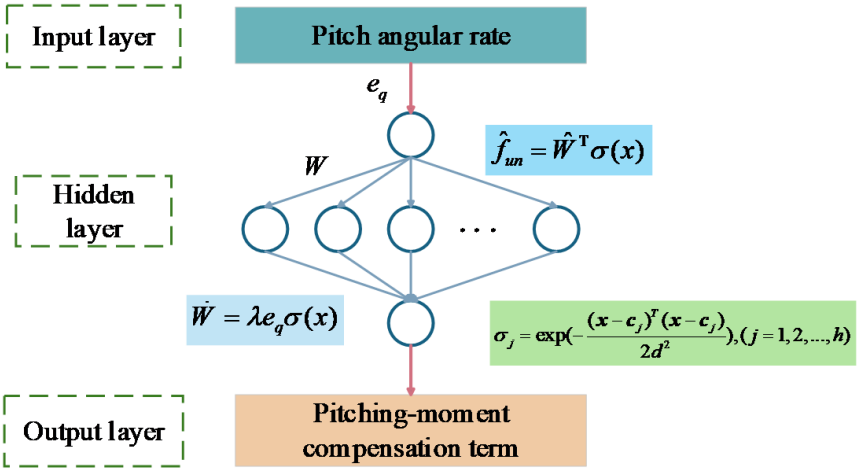


Figure 5. RBF network with pitch rate input and equivalent pitching moment correction.

Gaussian basis functions are used to represent the dependence on the selected input features within the assumed operating region. A basis function is most strongly activated when its input is close to its center and decays as the distance increases. Each center vector $\mathbf{c}_j$ has the same dimension as the network input $\mathbf{x}$. The hidden-layer output is defined as:

$$\sigma_j = \exp\left(-\frac{(\mathbf{x}-\mathbf{c}_j)^T(\mathbf{x}-\mathbf{c}_j)}{2d^2}\right), \qquad j = 1, 2, \ldots, h. \tag{32}$$

In the single-input implementation, $x = q$ denotes the pitch rate input coordinate. The Gaussian expression retains vector notation for generality, with dimension one in this study. The hidden-layer activation is $\sigma_j \in [0, 1]$; c_j is a scalar center, h is the number of Gaussian basis functions, and $d > 0$ is their common width. Input, center, and width values must use the same physical units or the same normalized coordinates. For basis function evaluation, pitch rate is converted from rad/s to °/s using $q_{\deg} = 180q/\pi$, without further normalization. The centers are uniformly spaced at 3°/s, from −80°/s to 79°/s, and share a common width. Centers and widths remain fixed; only the output-layer weights are updated online. Table 1 summarizes the basis count, center-generating rule, and width for the Gaussian approximation [42].

Table 1. Controller and actuator parameters.

Parameter	Value
Outer-loop PID (K_P, K_I, K_D)	(3, 5, 0.2)
Inner-loop PID (K_P, K_I, K_D)	(2, 1, 0.1)
RBF adaptation gain γ	0.2
RBF basis count h	54
RBF centers c_j (°/s)	$-80 + 3(j-1), j = 1, \ldots, 54$
RBF width d (°/s)	10
PID elevator-command limits (rad)	±15/57.3 (approximately ±15°)
Elevator rate limit (°/s)	±80
Controller period T_s (s)	0.01

The online output of the RBF neural network is expressed as

$$y(x) = \hat{W}^T\sigma(x), \tag{33}$$

where $y(x)$ denotes the network output, $\sigma(x)$ is the hidden-layer output vector composed of Gaussian basis functions, and $\hat{W}$ is the output-layer weight vector updated online.

Only the output-layer weights are adapted online; the basis centers and widths remain fixed. The analysis is local to a compact operating region Ω, where the reference and known model terms are bounded and ideal weights exist in the projection set. The residual bound is specified below. Projection bounds the estimated weights, but does not establish that all aircraft states remain in Ω.

Over the compact set Ω, the actual lumped disturbance entering the pitch-rate channel is expressed as

$$f_{\text{un}}(x,t) = W^{*T}\sigma(x) + \varepsilon(x,t), \tag{34}$$

where x is the input vector of the RBF neural network, W^* denotes the ideal constant weight vector over the compact set Ω and is introduced only for stability analysis, and $\varepsilon(x,t)$ is the RBF approximation error. The assumption that W^* is constant does not imply that the lumped disturbance is a constant offset, since the basis function vector $\sigma(x)$ varies with the network input along the system trajectory. Thus, $W^{*T}\sigma(x)$ represents the state-dependent component of the lumped disturbance that can be approximated using the selected network inputs. The residual $\varepsilon(x,t)$ includes both approximation inaccuracies and time-varying disturbances or unmodeled dynamics that cannot be uniquely represented by those inputs. In particular, an external disturbance may take different values at different times even when the network input is identical; such explicit time dependence is included in the residual. The subsequent stability analysis assumes that this residual is uniformly bounded over the considered operating region and time interval. Therefore, the disturbance representation accommodates bounded time-varying components without requiring exact reconstruction of arbitrary time-varying disturbances by the RBF network.

To achieve online compensation, an adaptive law for the output-layer weights is designed based on Lyapunov stability analysis. Unlike offline error backpropagation training, the proposed adaptive law updates the output-layer weights online while keeping the basis functions fixed, enabling the network to track the component of the lumped disturbance associated with the system state.

Considering the combined effects of external disturbances and model uncertainties on the dynamic inversion model, the pitch rate tracking error is defined as $e_q = q - q_c$. By incorporating $-\dot{q}_c$ into the known term g_q, the tracking error dynamics can be written as follows:

$$\dot{e}_q = f_q u + g_q + f_{\text{un}}(x,t). \tag{35}$$

For physical actuator input u, let f_q denote its effectiveness in the pitch angular acceleration equation. The inversion is restricted to the admissible operating region in which

$$0 < f_{\min} \le |f_q(x)| \le f_{\max} < \infty, \qquad x \in \Omega. \tag{36}$$

The input direction must be known. In a local elevator representation, $f_q = \bar{q}SlC_m^{\delta_e}/I_y$, and the moment command in Equation (18) must be realizable through the actuator map in Equation (19). An estimated effectiveness $\hat{f}_q$ must also remain nonzero with the correct input direction; the mismatch $(f_q - \hat{f}_q)u$ is included in the bounded residual. Equation (36) excludes near-zero effectiveness and sign reversal; identification does not restore lost physical control authority. Under these conditions, the ideal input command incorporating the RBF estimate is

$$u = -f_q^{-1}\left[ke_q + g_q + \hat{f}_{\text{un}}(x)\right]. \tag{37}$$

Let $\gamma > 0$ denote the weight adaptation gain, and define the weight estimation error as $\tilde{W} = \hat{W} - W^*$. Assuming that the ideal weight vector W^* is constant, the Lyapunov function is constructed as

$$V = \frac{1}{2}e_q^2 + \frac{1}{2\gamma}\tilde{W}^T\tilde{W}, \tag{38}$$

and substituting the control law in Equation (37) into the tracking error dynamics in Equation (35) yields

$$\begin{aligned} \dot{e}_q &= -f_q f_q^{-1}\left[ke_q + g_q + \hat{f}_{\text{un}}(x)\right] + g_q + f_{\text{un}}(x,t) \\ &= -ke_q + f_{\text{un}}(x,t) - \hat{f}_{\text{un}}(x). \end{aligned} \tag{39}$$

Thus,

$$\dot{e}_q = -ke_q - \tilde{W}^T\sigma(x) + \varepsilon(x,t). \tag{40}$$

Taking the time derivative of the Lyapunov function in Equation (38) (noting that $\dot{\tilde{W}} = \dot{\hat{W}}$ because the ideal weight vector W^* is assumed to be constant) then gives

$$\dot{V} = e_q\dot{e}_q + \frac{1}{\gamma}\tilde{W}^T\dot{\tilde{W}}. \tag{41}$$

Substituting the closed-loop tracking error dynamics into the above expression yields

$$\begin{aligned} \dot{V} &= -ke_q^2 - e_q\tilde{W}^T\sigma(x) + e_q\varepsilon(x,t) + \frac{1}{\gamma}\tilde{W}^T\dot{\hat{W}} \\ &= -ke_q^2 + e_q\varepsilon(x,t) - \tilde{W}^T\left[e_q\sigma(x) - \frac{1}{\gamma}\dot{\hat{W}}\right]. \end{aligned} \tag{42}$$

To ensure that the online weights remain bounded, the following projection-based RBF weight adaptation law is employed:

$$\dot{\hat{W}} = \text{Proj}_{\mathcal{W}}\left(\hat{W}, \gamma e_q\sigma(x)\right) \tag{43}$$

where $\mathcal{W} = \{W \in R^h : \|W\| \le W_{\max}\}$ is a known compact and convex set satisfying $W^* \in \mathcal{W}$ and the initial weight estimate satisfies $\hat{W}(0) \in \mathcal{W}$. The projection operator is defined as

$$\text{Proj}_{\mathcal{W}}(\hat{W}, y) = \begin{cases} y, & \|\hat{W}\| < W_{\max} \text{ or } \hat{W}^T y \le 0, \\ y - \dfrac{\hat{W}\hat{W}^T}{\|\hat{W}\|^2}y, & \|\hat{W}\| = W_{\max} \text{ and } \hat{W}^T y > 0. \end{cases} \tag{44}$$

This projection operator guarantees that $\hat{W}(t) \in \mathcal{W}$ and satisfies

$$\tilde{W}^T\left[\text{Proj}_{\mathcal{W}}(\hat{W}, y) - y\right] \le 0. \tag{45}$$

Based on the projection property in Equation (45), substituting Equation (43) into Equation (42) yields

$$\dot{V} \le -ke_q^2 + e_q\varepsilon(x,t). \tag{46}$$

The residual is assumed to be uniformly bounded on Ω over the time interval to which the stability result applies:

$$|\varepsilon(x,t)| \le \bar{\varepsilon}. \tag{47}$$

To derive an explicit ultimate bound for the tracking error, assume that the RBF basis function vector is bounded as

$$\|\sigma(x)\| \le \bar{\sigma}. \tag{48}$$

Since the projection operator guarantees $\hat{W}(t) \in \mathcal{W}$, the weight estimation error satisfies

$$\|\tilde{W}\| = \|\hat{W} - W^*\| \le \bar{W}_\Delta, \tag{49}$$

where

$$\bar{W}_\Delta = \sup\|\hat{W} - W^*\|, \qquad \hat{W} \in \mathcal{W}. \tag{50}$$

Define the upper bound of the lumped residual term as

$$D = \bar{W}_\Delta\bar{\sigma} + \bar{\varepsilon}. \tag{51}$$

It then follows from Equation (40) and the bounds in Equations (47)–(51) that

$$D^+|e_q| \leq -k|e_q| + D, \tag{52}$$

where $D^+|e_q|$ denotes the upper-right Dini derivative of $|e_q|$. According to the comparison lemma,

$$|e_q(t)| \leq e^{-kt}|e_q(0)| + \frac{D}{k}(1 - e^{-kt}). \tag{53}$$

Provided that the trajectory remains in the admissible region and the stated bounds hold uniformly in time, the pitch rate tracking error is uniformly ultimately bounded. This is a conditional pitch channel result, not a global boundedness proof for all aircraft states. Its ultimate bound satisfies

$$\limsup_{t\to\infty} |e_q(t)| \leq \frac{D}{k}. \tag{54}$$

Equation (54) shows that increasing the control gain k or reducing the residual bound D decreases the ultimate bound of the tracking error. In the special case $\bar{\varepsilon} = 0$, Equations (46) and (47) give $\dot{V} \leqslant -ke_q^2$. By integrating this inequality over [0,T], using V(T) ≥ 0, and then letting $T \to \infty$, one obtains

$$\int_0^\infty e_q^2(t)\, dt \leq \frac{V(0)}{k} < \infty. \tag{55}$$

If all closed-loop signals are bounded and $\dot{e}_q$ is bounded, then e_q is uniformly continuous; therefore, Barbalat's lemma gives $e_q \to 0$. However, convergence of the estimated weight vector $\hat{W}$ to the ideal weight vector W^* additionally requires a persistent excitation condition.

Equation (54) also covers bounded time-varying residuals. The actuator realization error is included below.

Actuator Realization and the Conditional Error Bound

Let u_c be the ideal command from Equation (37) and u_a the input actually realized by the actuator, in the same physical units. The amplitude and rate constraints are

$$u_{\min} \leq u_a \leq u_{\max}, \qquad |\dot{u}_a| \leq r_{\max}. \tag{56}$$

The available elevator command amplitude limits are listed in Table 1. Substituting u_a into Equation (35) gives

$$\begin{aligned} \dot{e}_q &= -ke_q - \tilde{W}^T\sigma(x) + \varepsilon(x,t) + d_a(t), \\ d_a(t) &= f_q(x)\,[u_a(t) - u_c(t)]. \end{aligned} \tag{57}$$

Using the realized-input error dynamics in Equation (57) and the projected adaptation law in Equation (43), the derivative of the Lyapunov function in Equation (38) satisfies

$$\dot{V} \leq -ke_q^2 + e_q\,[\varepsilon(x,t) + d_a(t)]. \tag{58}$$

If an independently justified bound $|d_a(t)| \leq \bar{d}_a$ holds uniformly in the admissible region, applying the bounds in Equations (47)–(51) to Equation (57) yields

$$D^+|e_q| \leq -k|e_q| + D + \bar{d}_a, \tag{59}$$

and applying the comparison lemma to Equation (59) gives

$$\begin{aligned} |e_q(t)| &\leq e^{-kt}|e_q(0)| + \frac{D + \bar{d}_a}{k}(1 - e^{-kt}), \\ \limsup_{t\to\infty} |e_q(t)| &\leq \frac{D + \bar{d}_a}{k}. \end{aligned} \tag{60}$$

The bound in Equation (60) accounts for actuator lag and amplitude/rate constraints through d_a. It requires a bounded command mismatch and continued operation in Ω; however, bounded actuator output alone does not establish these conditions.

With the network weights updated online according to the adaptation law in Equation (43), the RBF neural network estimate of the actual lumped disturbance is given by

$$\hat{f}_{\text{un}}(x) = \hat{W}^T \sigma(x). \tag{61}$$

As shown in Figure 6, the RBF network augments the NDI controller with residual estimation and compensation. The offline-trained PINN supplies aerodynamic model information through online inference with fixed weights and biases. The RBF basis functions use measured pitch rate q, while the pitch rate tracking error e_q drives projected output-weight adaptation. The disturbance estimate is converted to an equivalent pitching moment correction before control allocation. Components that cannot be represented by the selected input remain in the bounded residual assumed in the analysis.

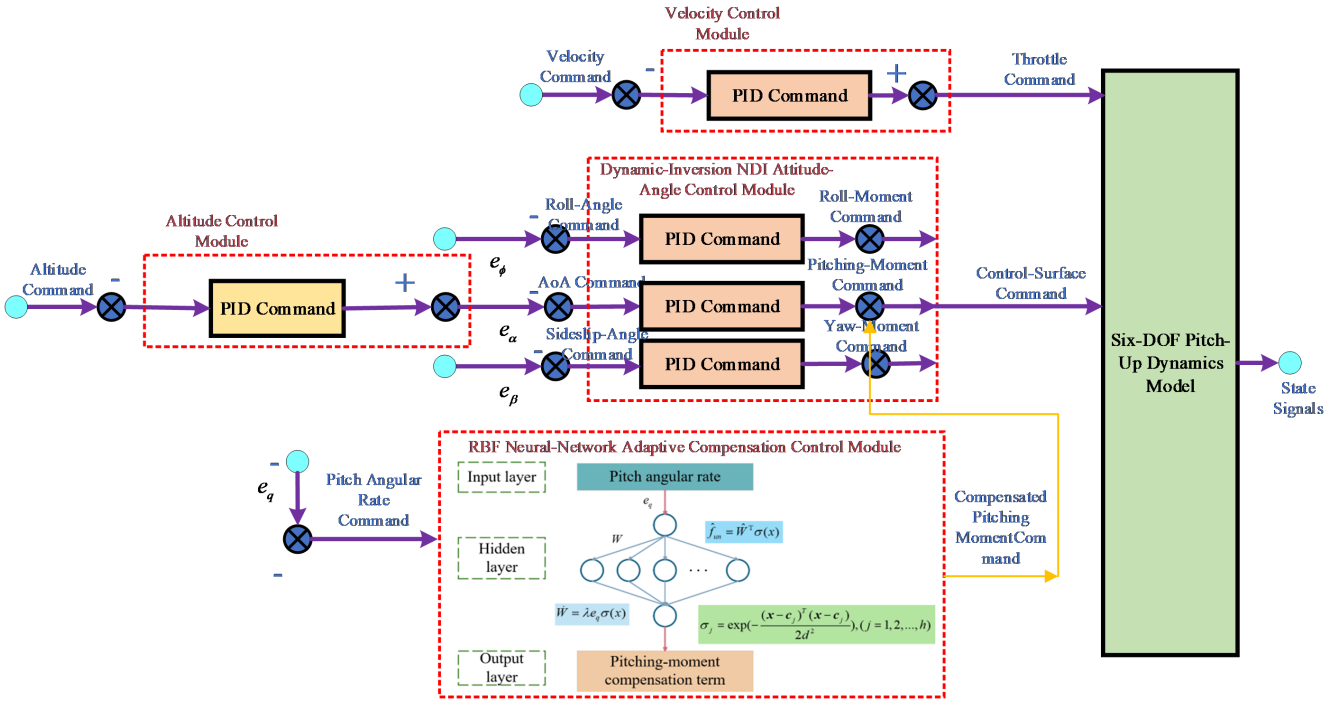


Figure 6. Compensation framework of the NDI control law based on the RBF neural network.

4. Simulation Comparison Analysis

The numerical evaluation uses the open-source F-16 benchmark to compare aerodynamic identification and longitudinal tracking. Performance is assessed through tracking-error magnitude, transient response, and actuator activity under prescribed angle-of-attack commands. These are numerical control performance measures; they do not quantify accident risk or validate loss-of-control prevention. The simulation process is shown in Figure 7.

The initial conditions are set as follows: altitude H = 1500 m, Mach number 0.8, and angle-of-attack step command during the first 5 s $\alpha = 8°$. After 5 s, the UAV enters the high-angle-of-attack condition and the angle-of-attack command is stepped to 15°. During the entire simulation, the velocity command is maintained at Mach 0.8. The simulation duration is 10 s. The simulation objective is to track the commanded angle of attack, regulate the UAV velocity through throttle control, and stabilize the angle of attack through elevator control.

The 8° to 15° command change is a prescribed tracking test; it does not represent an emergency stall recovery trajectory, a turbulence encounter, or a validated civil flight enve-

lope. The benchmark Mach number and altitude are retained for the numerical comparison, and are not proposed as operating conditions for a civil UAV.

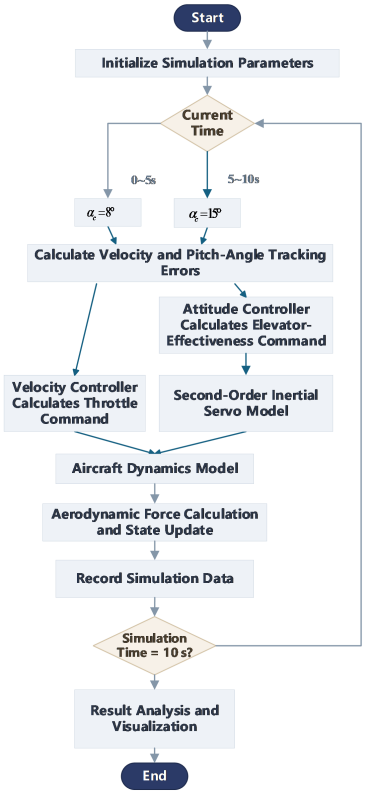


Figure 7. Simulation flowchart.

4.1. Controller Settings and Common Evaluation Protocol

The evaluated strategies are the implemented PID baseline, PINN + NDI, and PINN + NDI + RBF. The PID baseline uses a cascaded structure. The outer loop takes the angle-of-attack error as its input and generates the pitch rate command, while the inner loop takes the pitch rate error as its input and generates the elevator command. The PID gains were tuned manually by trial and error at an altitude of 1500 m and a Mach number of 0.8, using an angle-of-attack command change from 8° to 15°. The selection jointly considered angle-of-attack overshoot, settling time, tracking error, and elevator deflection, aiming for small tracking errors, limited oscillations, and reasonable control commands. The inner- and outer-loop PID gains listed in Table 1 remain fixed during evaluation. The controller sampling period is $T_s = 0.01$ s. Both PID loops use derivative filtering with $N = 100$. The outer loop uses clamping anti-windup and integrator output limits of $[-0.1, 0.1]$, with no finite internal PID output limits. The inner loop has no anti-windup, integrator output limiting, or internal PID output limiting.

The performance variable in the reported tracking figures is angle of attack. Under planar longitudinal motion, the pitch angle, angle of attack, and flight path angle satisfy $\vartheta = \alpha + \gamma$. Hence, $q = \dot{\alpha} + \dot{\gamma}$, where γ is the flight path angle. In the PID baseline, the outer-loop output provides the desired angle-of-attack rate $\dot{\alpha}_{\mathrm{des}}$. The flight path angle signal is differentiated to obtain $\dot{\gamma}$ (the Dgamma signal), which is added to the outer-loop output before the inner-loop error is formed. The resulting pitch rate reference is $q_c = \dot{\alpha}_{\mathrm{des}} + \dot{\gamma}$, and the inner-loop error is $e_q = q - q_c$. This kinematic relation connects angle-of-attack tracking to the pitch angle dynamics in Section 3.1, with flight path angle variation retained

in the reference calculation. Define $e_\alpha(t) = \alpha(t) - \alpha_c(t)$ on a common interval $\mathcal{I} = [t_0, t_f]$, with $T = t_f - t_0 > 0$. The tracking metrics are

$$\begin{aligned} E_{\mathrm{RMSE}} &= \sqrt{\frac{1}{T}\int_{t_0}^{t_f} e_\alpha^2(t)\,dt}, \\ E_{\mathrm{MAE}} &= \frac{1}{T}\int_{t_0}^{t_f} |e_\alpha(t)|\,dt, \\ E_{\mathrm{pk}} &= \max_{t\in\mathcal{I}} |e_\alpha(t)|. \end{aligned} \tag{62}$$

The full interval and the two command segments, $[0, 5)$ and $[5, 10]$ s, are reported separately. The recorded reference α_c and angle-of-attack responses are evaluated on their common 0.01 s grid. The 10–90% rise time t_r is the interval between the first 10% crossing and the first subsequent 90% crossing of the commanded increase. The initial measured angle of attack is used as the baseline for the first segment, and the preceding reference level is used for the second segment. Crossing times are resolved on the recorded grid. At an upward reference step, overshoot is $O_{\max} = \max(0, \max[\alpha(t) - \alpha_c(t)])$ over that segment. It is distinct from E_{pk}, which includes the tracking error introduced by the reference jump. The settling time tolerance is 0.14°; settling time is measured from the segment start until the response enters and remains within this band for the rest of the segment. A response that does not settle within the observation interval is reported as such rather than being assigned a settling time of zero.

Control activity is measured using the realized elevator deflection:

$$J_\delta = \int_{t_0}^{t_f} \delta_{e,a}^2(t)\,dt. \tag{63}$$

The quantity J_δ in Equation (63) measures actuator activity rather than mechanical energy. Maximum absolute deflection, maximum absolute deflection rate, and saturation duration are also reported. The integration and common resampling method is segment-wise trapezoidal integration on the shared 0.01 s grid, without resampling.

4.2. Aerodynamic Parameter Identification Simulation Based on PINN

To improve the aerodynamic model used in the high-angle-of-attack simulations, a PINN is trained offline using model-generated data. Its inputs are angle of attack, pitch rate, and elevator deflection, while its output is the pitching moment coefficient. The data cover $\alpha \in [-4, 17]^\circ$, $\delta_e \in [-24, 6]^\circ$, and $q \in [0, 23]^\circ/\mathrm{s}$. These are data generation ranges, not established actuator limits or a positively invariant closed-loop region.

Reference pitching moment coefficients provide supervision targets for the data loss, while the physics loss uses sampled pitch rates and network-predicted coefficients as specified in Equation (29). The training and test partitions are formed before preprocessing or adjacent-sample pairing. Normalization parameters are obtained from the training partition only, and the test partition is reserved for final evaluation. A physics loss pair is formed only from time-adjacent samples of the same trajectory within the same partition; no pair crosses a train/test boundary or joins independent operating point records. The normalization map and its inverse are $z_{\mathrm{n}} = (z - z_{\min})/(z_{\max} - z_{\min})$ and $z = z_{\mathrm{n}}(z_{\max} - z_{\min}) + z_{\min}$, respectively, for each normalized input or output variable z. The supervised reference coefficients of test samples do not enter the training loss or parameter updates.

After training, the PINN weights and biases are fixed. During closed-loop operation, the network is evaluated at the current inputs to obtain the pitching moment coefficient. Aerodynamic derivatives are calculated using forward finite differences of this fixed network output with respect to the relevant input coordinates. The perturbation sizes for angle of attack and elevator deflection are 0.01 rad and 0.01 rad, respectively. These input-space

differences are separate from the Euler time update in Equation (29); they do not numerically differentiate the sampled pitch rate time series. The coefficient and derivative estimates are recalculated when the inputs change and update the NDI model and control effectiveness terms. No online PINN weight optimization or retraining is performed. Figure 8 shows the offline training loss history, and Table 2 summarizes the network architecture and offline training settings.

The 3–4–8–4–1 network contains 97 trainable parameters. Online inference was timed on an Intel Core i7-14700KF CPU (3.40 GHz) using Python 3.14.7 and PyTorch 2.13.0+cpu; the installed NVIDIA GeForce GT 1030 GPU (4 GB) was not used. With a batch size of one, the mean and largest observed forward-pass latencies in the same test were 0.028386 and 0.2489 ms, respectively, compared with the 10 ms controller sampling period. These timings exclude offline training and aerodynamic-derivative calculations.

Table 2. PINN architecture and offline training settings.

Name	Value
Network structure	3–4–8–4–1
Network learning rate η	0.05
Training set	90%
Test set	10%
Training target accuracy $\mathcal{L}_{target}$	10^{-6}
Maximum training epochs	50,000
Weight of data loss λ_1	0.9
Weight of physics loss λ_2	0.1

As shown in Figure 8, the physics-informed neural network exhibits stable convergence characteristics. The data loss term and physics residual term show a synchronous convergence trend, indicating that the embedded physical equations provide a positive regularizing effect on the gradient descent direction.

The identification results obtained during the simulation are compared with the true aerodynamic parameters, and the corresponding simulation results are shown in Figures 9 and 10.

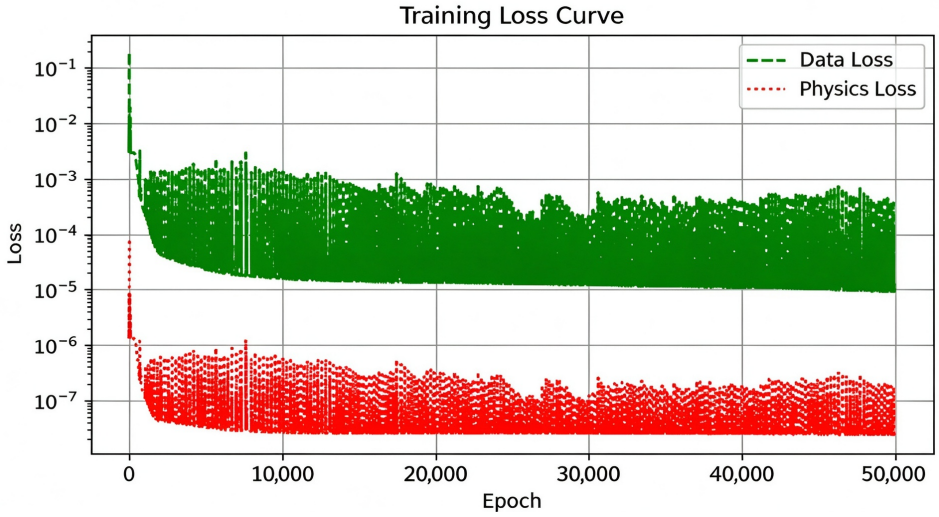


Figure 8. Loss function convergence curve of the PINN.

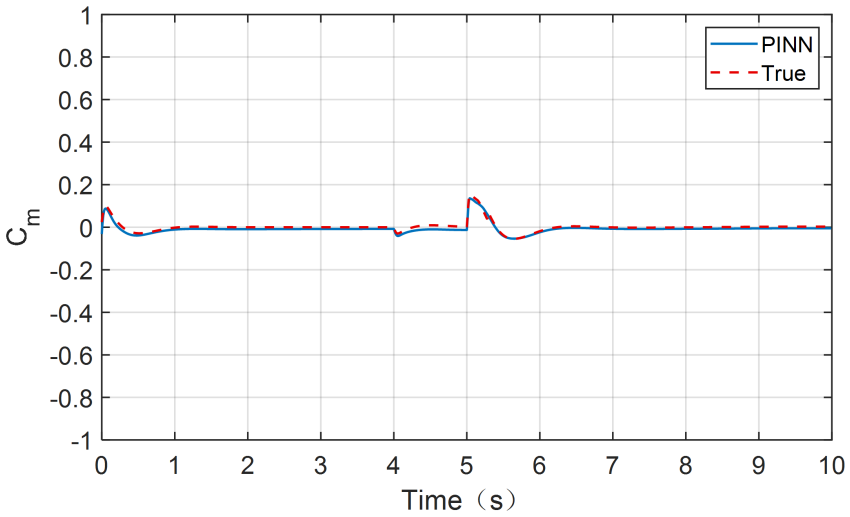
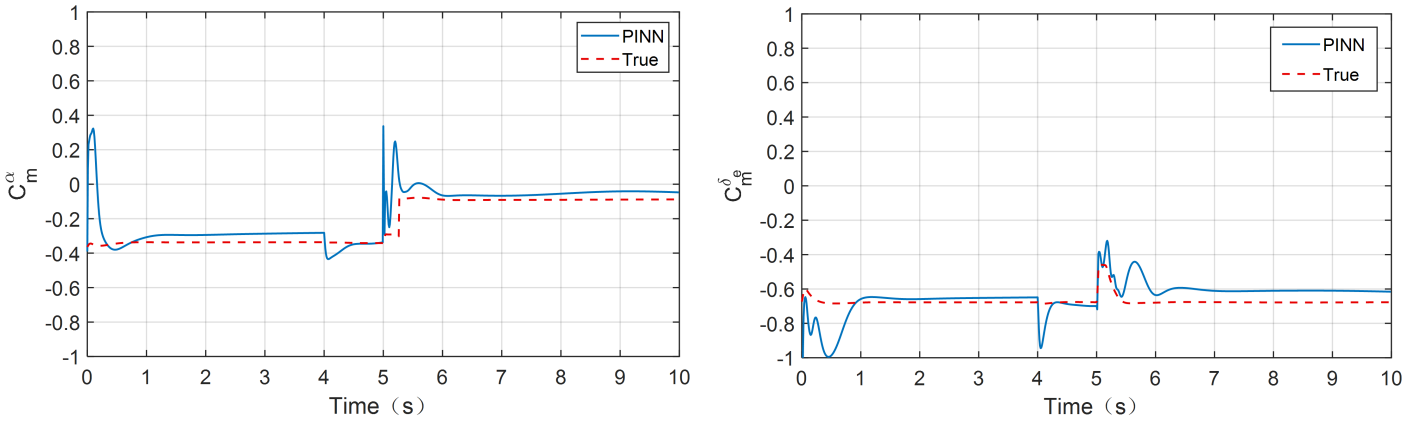


Figure 9. Comparison curve of the pitching moment coefficient.



(a) **(b)**

Figure 10. Identification curves of the pitching moment coefficient derivatives: (**a**) static stability coefficient and (**b**) elevator effectiveness.

For the identification results of the pitching moment coefficient C_m, the PINN prediction curve is highly consistent with the true value. The mean squared error (MSE) is 0.0001, indicating that the model can accurately track the initial response, the peak value of the pitching moment coefficient after the angle-of-attack step at t = 5 s, and the attenuation process. The predicted results remain in good agreement with the true values in terms of variation trend, response amplitude, and phase, with only slight deviations near local peaks. This indicates that the PINN can effectively learn the nonlinear dynamic characteristics of the pitching moment coefficient.

As indicated by the identification curves of the pitching-moment coefficient derivatives, the PINN can correctly capture the signs, steady-state ranges, and overall variation trends of the static stability coefficient C_m^α and elevator effectiveness $C_m^{\delta_e}$. Under steady operating conditions, the predicted values of the two derivatives are close to the true values, with MSEs of 0.0110 and 0.0085, respectively. During the initial stage and the operating-condition transition caused by the angle-of-attack step, the predicted curves exhibit noticeable overshoot and attenuated oscillations, but they subsequently reconverge. Compared with the identification of C_m, the local deviations of the aerodynamic derivatives are more pronounced, mainly because differential calculation amplifies small prediction errors in the model output.

Overall, the PINN achieves high identification accuracy for the pitching moment coefficient and effectively captures the dynamic variations caused by operating-condition transitions. It also provides good trend identification capability for the static stability coefficient and elevator effectiveness.

4.3. High-Angle-of-Attack Longitudinal Control Simulation Based on PINN + NDI

The PINN + NDI simulations use the elevator effectiveness identification results from Section 4.2 to evaluate tracking in the prescribed benchmark. The comparison is relative to the implemented PID baseline with the settings and common metrics defined above; it is not a comparison against all PID-based methods.

A second-order actuator model represents actuator lag. The plant elevator effectiveness setting is held constant and nonzero within each simulation; the ±20% perturbations in Appendix A are likewise fixed within each case. Elevator deflection and deflection rate are constrained by the limits in Table 1, and the recorded responses remain within these limits. No explicit limits are imposed on angle of attack or pitch rate, for which the responses remain bounded within the operating range of the prescribed test over the simulated interval. These observations support the operating conditions considered in the analysis without establishing its residual bound or extending the result to near-zero effectiveness. Figure 11 shows the PID and PINN + NDI responses.

Figure 11 shows the responses to the initial command and the step at $t = 5$ s. Within the displayed test, PINN + NDI has a smaller plotted overshoot than the implemented PID baseline. The elevator, Mach number, altitude, and throttle curves describe the associated transient responses. Quantitative tracking and actuator metrics are reported in Table 3; broader claims about settling time, control effort, or robustness are limited to the completed entries and tested conditions.

Table 3. Common tracking and control activity metrics. Transient peak deviation and peak absolute tracking error are distinct measures.

Strategy	Interval (s)	RMSE (°)	MAE (°)	E_{pk} (°)	$O_{\max}$ (°)	t_r (s)	t_s (s)	J_δ (°^{2}s)
PID	[0, 5)	2.6660	1.3940	9.1510	1.3617	0.59	3.35	183.2035
PINN + NDI	[0, 5)	2.4158	1.2445	8.7150	1.0114	0.75	3.57	183.1796
PINN + NDI + RBF	[0, 5)	2.2442	1.0899	8.5854	0.6199	0.78	3.85	182.8447
PID	[5, 10]	2.2083	1.2208	7.3643	1.5479	0.46	4.63	282.0681
PINN + NDI	[5, 10]	1.7733	0.8556	7.0433	0.7694	0.71	2.78	281.2401
PINN + NDI + RBF	[5, 10]	1.5680	0.6537	6.9706	0.3302	0.73	2.60	281.2200

The supplementary pitch-up suppression index compares the reported transient peak deviations under two controllers. It does not replace the RMSE, MAE, peak absolute tracking error, settling time, and actuator metrics defined above. With $\Delta\alpha_{\mathrm{PID,max}}$ and $\Delta\alpha_{\mathrm{NDI,max}}$ denoting the same selected transient deviation measure under the PID and PINN + NDI strategies, respectively, the index is

$$\eta = \left(1 - \frac{\Delta\alpha_{\mathrm{NDI,max}}}{\Delta\alpha_{\mathrm{PID,max}}}\right) \times 100\%. \tag{64}$$

A larger η indicates a greater reduction of the selected transient peak deviation relative to the PID baseline. The source results report approximately 1.5479° and 0.7694°, giving a reduction of approximately 50.29%. These values are retained as reported transient deviation measures, and are not identified with the full-interval E_{pk}; the reference jump itself introduces a larger immediate tracking error. The precise transient window and definition used for the reported deviations are [5, 10] s and $\Delta\alpha_{\max} = O_{\max}$, respectively. The index does not quantify accident probability, safety improvement, or a certified stability margin.

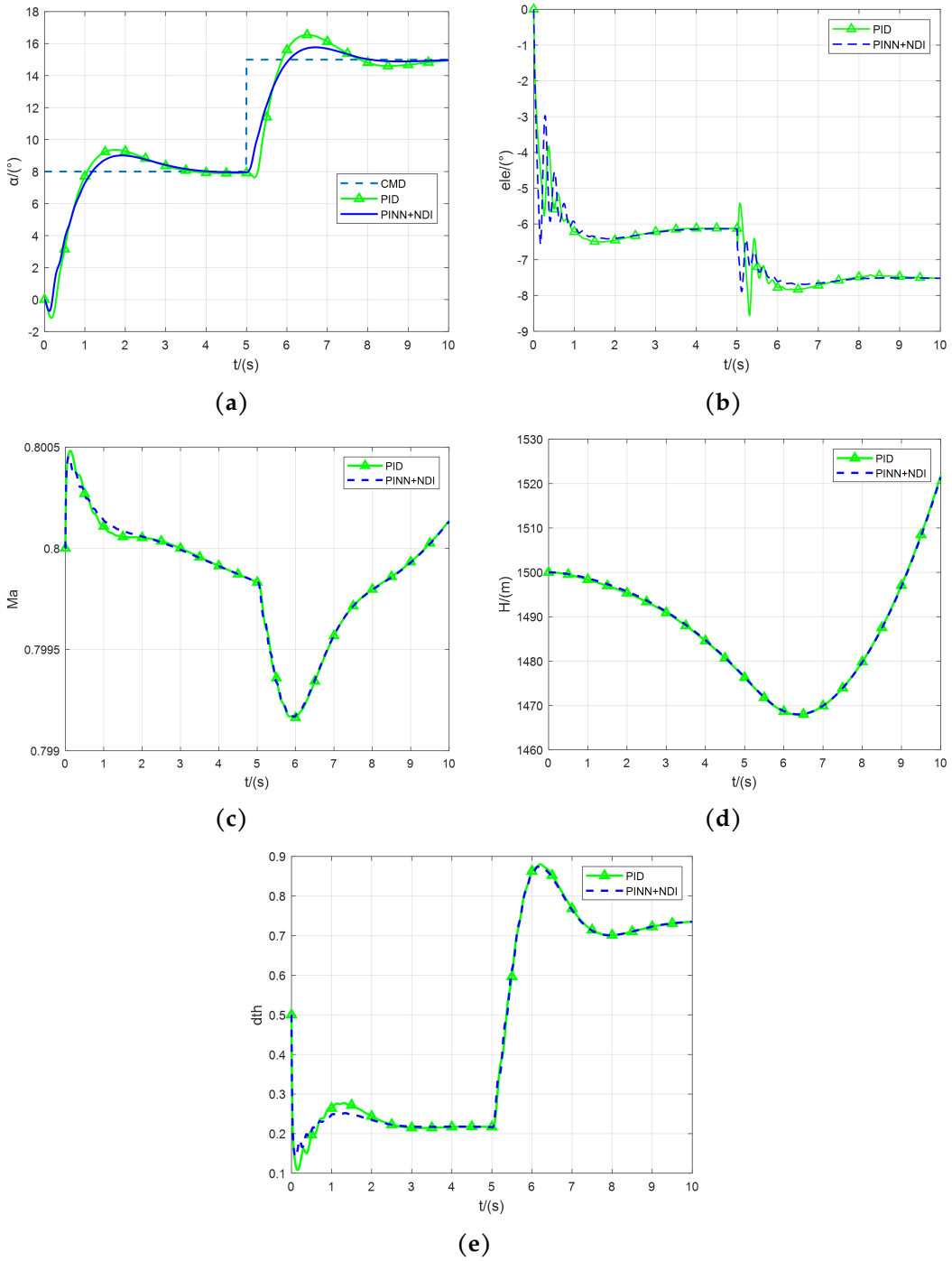


Figure 11. Comparative simulation results of high-angle-of-attack tracking control: (**a**) angle-of-attack control comparison curve, (**b**) elevator deflection comparison curve, (**c**) velocity comparison curve, (**d**) altitude comparison curve, and (**e**) throttle comparison curve.

4.4. High-Angle-of-Attack Longitudinal Control Simulation with RBF Compensation

The RBF-compensated simulation compares the PID baseline, PINN + NDI, and PINN + NDI + RBF under the prescribed numerical conditions. The PINN supplies model information and the RBF term compensates for remaining pitch channel uncertainty. Figure 12 shows tracking and coupled-state responses. These responses support a numerical tracking comparison; a reference step or a model mismatch alone is not an explicit validation against arbitrary time-varying external disturbances.

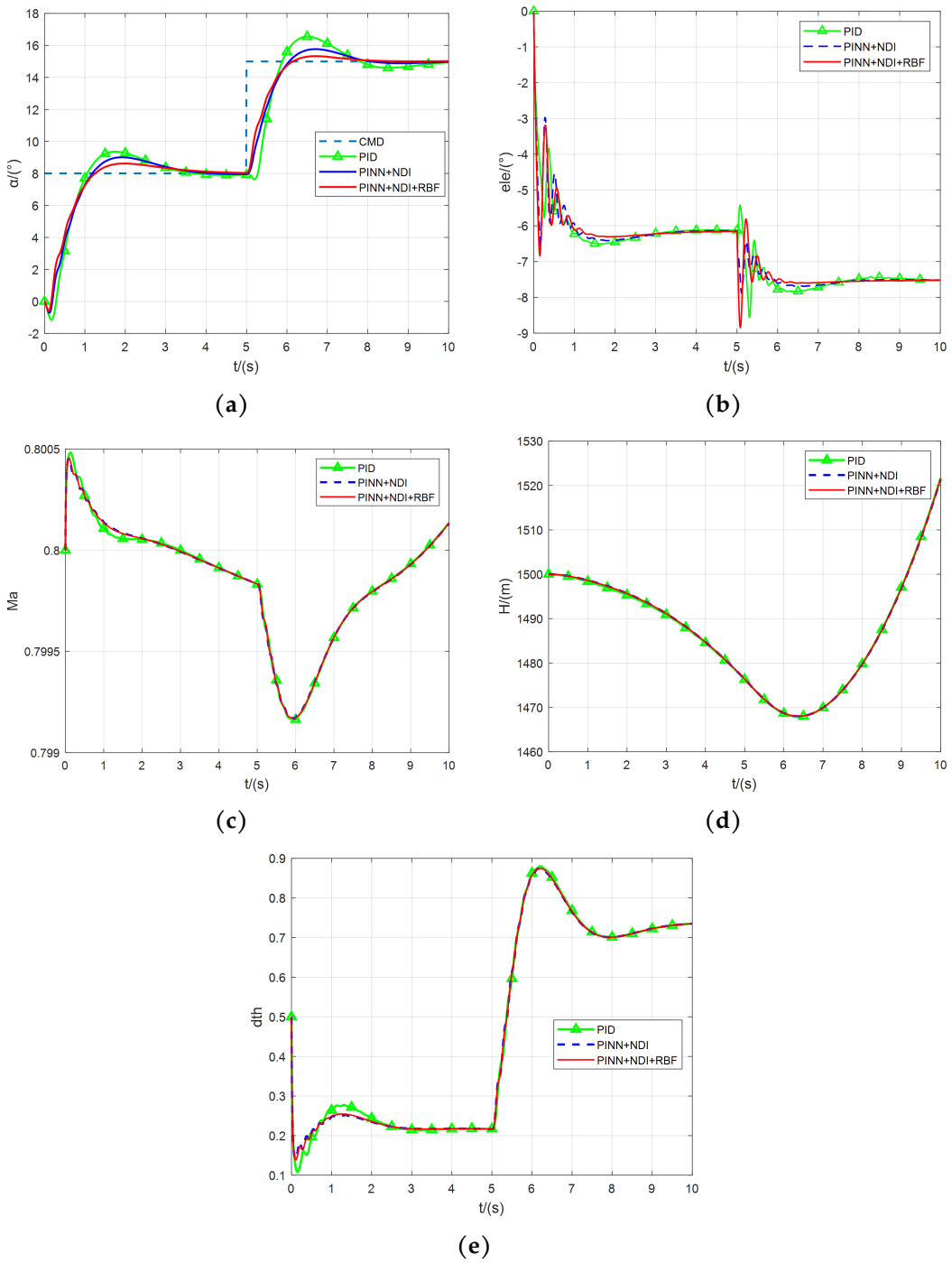


Figure 12. Comparative simulation results with RBF compensation: (**a**) angle-of-attack control comparison curve, (**b**) elevator deflection comparison curve, (**c**) velocity comparison curve, (**d**) altitude comparison curve, and (**e**) throttle comparison curve.

In Figure 12, RBF compensation reduces the plotted transient peak deviation relative to PINN + NDI in the displayed case. The elevator response also shows a changed transient control action. Whether this trade-off reduces integrated control activity or settling time is determined by the metrics in Table 3. A smoother plotted response alone does not establish lower actuator effort or robustness over untested conditions.

The displayed Mach number, altitude, and throttle responses are broadly similar across the evaluated strategies in this test. This observation is restricted to the plotted trajectories and is not a general guarantee of unaffected flight state regulation or energy management.

The source results report a transient peak deviation of approximately 0.3302° with RBF compensation, compared with approximately 1.5479° for PID, giving a supplementary reduction ratio of approximately 78.67%. The associated interval and transient deviation definition must be the same as those specified for Equation (64). The full tracking error metrics are reported separately.

The displayed responses indicate smaller transient deviations for the model-informed and RBF-compensated strategies in the prescribed test. Table 3 provides the common quantitative comparison, Table 4 reports realized elevator activity and constraint checks, and Table 5 summarizes the tracking errors over the common evaluation interval. Conclusions about performance improvement are restricted to those data and the implemented PID baseline.

Table 4. Realized elevator activity and constraint checks over the common evaluation interval.

Strategy	$\max \lvert\delta_{e,a}\rvert$ (°)	$\max \lvert\dot{\delta}_{e,a}\rvert$ (°/s)	Saturation Time (s)
PID	8.5614	75.0394	NR
PINN + NDI	7.8868	75.0394	NR
PINN + NDI + RBF	8.8454	73.3206	NR

Deflection rates are estimated from successive samples at 0.01 s intervals. NR: saturation duration cannot be established from realized deflection logs alone. The PID elevator command limits are approximately ±15°; limiter input signals or saturation flags are required, together with the applicable limits for each strategy.

Table 5. Tracking error summary over $[0, 10]$ s.

Strategy	RMSE (°)	MAE (°)
PID	2.4479	1.3074
PINN + NDI	2.1190	1.0501
PINN + NDI + RBF	1.9359	0.8718

Quantitative Tracking and Actuator Comparison

For each controller, the metrics in Equations (62) and (63) are calculated from the same reference and time windows. The parameter identification MSE values measure model accuracy and are reported separately from the tracking RMSE and MAE. The common error and actuator metrics are summarized below.

Additional simulations were conducted with the elevator effectiveness contribution perturbed by +20% and −20% relative to its nominal value. In both cases, PINN + NDI + RBF maintained stable tracking of the prescribed 8–15° angle-of-attack command, with limited transient deviations and a small steady-state error, satisfying the tracking objective. The perturbation settings and angle-of-attack responses are presented in Appendix A. Within each run, the same initial condition, parameter set, and disturbance/noise realization are used for all controllers. If m_r is a run-level RMSE, MAE, or control activity metric, then the across-run summaries are

$$\begin{aligned} \bar{m} &= \frac{1}{N}\sum_{r=1}^{N} m_r, \\ s_m &= \sqrt{\frac{1}{N-1}\sum_{r=1}^{N}(m_r - \bar{m})^2}, \qquad N > 1. \end{aligned} \tag{65}$$

Equation (65) defines the mean and standard deviation across independent simulation realizations. Time samples from a trajectory are not independent realizations. A fixed scenario set describes variation across those scenarios, and is not automatically a random population

sample. The relative reduction for a nonzero baseline metric is $100(\bar{m}_{\mathrm{PID}} - \bar{m}_{\mathrm{method}})/\bar{m}_{\mathrm{PID}}$; the same aggregation convention is used in the conclusions.

Over the common interval $[0, 10]$ s, the RMSE values for PID, PINN + NDI, and PINN + NDI + RBF are 2.4479, 2.1190, and 1.9359°, respectively, while the MAE values are 1.3074, 1.0501, and 0.8718°. The corresponding control activity values are 465.2716, 464.4197, and 464.0647°^{2} s. The tracking error statistics are interpreted together with the peak and settling metrics and the actuator activity when assessing the performance trade-off.

4.5. Scope and Limitations of Numerical Validation

The demonstrated evidence concerns the specified F-16 numerical model and the prescribed tracking commands. Lower plotted overshoot or lower completed error metrics describe numerical controller performance; they do not quantify accident risk reduction, establish a certified margin, or demonstrate flight safety on a civil UAV.

Dedicated stall recovery trajectories, severe turbulence, complete loss of control effectiveness, and real aircraft safety outcomes have not been validated. The result in Section 3.3 is conditional on its admissible region, residual, control effectiveness, and actuator realization assumptions. Future application requires representative aircraft models, explicit disturbance and uncertainty scenarios, feasible actuator authority, hardware-in-the-loop assessment, and appropriately controlled flight tests. These are separate validation steps rather than consequences of the present tracking plots.

5. Conclusions

This study evaluated a PINN-informed NDI controller with RBF residual compensation using an open-source F-16 numerical benchmark. The reported parameter identification MSE values for the pitching moment coefficient, static stability coefficient, and elevator effectiveness were 0.0001, 0.0110, and 0.0085. These identification errors are distinct from the closed-loop tracking error measures.

For the common evaluation interval $[0, 10]$ s, the angle-of-attack tracking RMSE values of the PID baseline, PINN + NDI, and PINN + NDI + RBF were 2.4479, 2.1190, and 1.9359°, respectively, while their MAE values were 1.3074, 1.0501, and 0.8718°. Relative to the implemented PID baseline, the RMSE and MAE reductions of PINN + NDI + RBF were 20.92% and 33.32%, using the same data interval and aggregation rule as Tables 3 and 5. The control activity measure changed from 465.2716 to 464.0647°^{2} s. While these results characterize the prescribed simulation conditions, they do not establish statistical significance.

The previously reported transient peak deviations of approximately 1.5479°, 0.7694°, and 0.3302° correspond to supplementary reduction ratios of approximately 50.29% and 78.67% relative to PID. They are not RMSE values, complete tracking error statistics, or measures of safety improvement. Any improvement in tracking must be interpreted together with the actuator activity and feasibility checks.

The stability analysis provides a conditional pitch rate error bound under bounded residual uncertainty, an admissible operating region, effective control authority, and bounded input realization error. It does not establish global aircraft stability near a loss of effectiveness or under arbitrary saturation. The numerical evidence is restricted to the tested F-16 benchmark and the implemented PID baseline, and does not establish superiority over hybrid, fuzzy, adaptive, or neural network-assisted PID controllers. Real aircraft flight safety, loss-of-control prevention, emergency stall recovery, and severe turbulence performance remain unvalidated and require dedicated future investigation.

Author Contributions: Conceptualization, D.Z.; methodology, D.Z.; software, D.Z. and B.H.; investigation, D.Z. and Y.W.; data curation, D. Z. and S.H.; writing—original draft preparation, D.Z. and Y.W.; writing—review and editing, D.Z. and K.L.; visualization, D.Z. and S.H.; funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Supported by the Fundamental Research Funds for the Central Universities, DUT26RC(3)119.

Data Availability Statement: The data are protected by confidentiality agreements and cannot be shared publicly.

DURC Statement: This study concerns longitudinal flight stability regulation, with civil aviation safety and loss-of-control prevention as the intended applications. General-purpose UAV control methods have dual-use potential. The F-16 model's military origin is acknowledged; however, it is used here solely as a numerical dynamics benchmark. The study does not investigate weapons integration, targeting, engagement, or offensive mission performance. The intended use of the reported findings is non-offensive flight safety research. The scope of the simulations and the need for application-specific safety validation are stated explicitly. Any subsequent application requires an appropriate safety assessment and compliance with applicable laws and aviation regulations.

Conflicts of Interest: Authors Yeguang Wang were employed by the Shengyang Aircraft Design and Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Angle-of-Attack Tracking Under Elevator Effectiveness Perturbations

The elevator effectiveness contribution in the aircraft model was perturbed by ±20% relative to its nominal value. The perturbation was applied to the `Cm2` branch associated with the elevator-induced pitching moment coefficient prior to the existing Mach number-dependent processing. The nominal branch gain of 1.5 was changed to 1.8 for the +20% case and to 1.2 for the −20% case.

Figures A1 and A2 show the angle-of-attack responses of PID, PINN + NDI, and PINN + NDI + RBF. The reference is 8° during the first 5 s and 15° thereafter, over a 10 s simulation interval. In both perturbation cases, the proposed PINN + NDI + RBF controller tracks the reference with a small steady-state error and lower transient overshoot than the PID baseline. The responses settle toward the commanded values without sustained oscillation, supporting the tracking objective under the two specified effectiveness perturbations. These comparisons assess closed-loop tracking performance under the prescribed parameter uncertainty.

The quantitative comparison follows the definitions in the evaluation protocol reported in the main text. Metrics are calculated from the recorded angle of attack and reference on the common 0.01 s grid, separately over $[0, 5)$ and $[5, 10]$ s and over the full $[0, 10]$ s interval. The overshoot $O_{\max}$ is the maximum positive tracking error in each segment. The rise time t_r is the 10–90% crossing interval, using the measured initial angle of attack for the first command and the preceding reference level for the second command. The settling time t_s is measured from the segment start until all subsequent recorded samples in that segment remain within ±0.14° of the reference. Thus, settling times for the second segment are measured relative to $t = 5$ s. RMSE and MAE use the same segment-wise trapezoidal integration convention as the results in the main text.

For the +20% case (Table A1), the overshoots following the command change at $t = 5$ s are 1.3784°, 1.1548°, and 0.8752° for PID, PINN + NDI, and PINN + NDI + RBF, respectively; the corresponding rise times are 0.41, 0.32, and 0.30 s, and the settling times are 3.67, 2.60, and 2.83 s. The RBF-compensated controller reduces the second-segment overshoot by 36.51% relative to PID. Over $[0, 10]$ s, its RMSE and MAE are 1.5418° and 0.6306°, which are 22.62% and 31.21% lower than the PID values. The absolute tracking error at $t = 10$ s is 0.0042°.

Table A1. Angle-of-attack tracking metrics under a +20% elevator effectiveness perturbation.

Strategy	Interval (s)	RMSE (°)	MAE (°)	$O_{\max}$ (°)	t_r (s)	t_s (s)
PID	$[0, 5)$	2.1646	0.9620	1.0332	0.51	2.67
PINN + NDI	$[0, 5)$	1.9090	0.7882	0.9148	0.35	2.84
PINN + NDI + RBF	$[0, 5)$	1.6739	0.6677	0.7346	0.32	2.95
PID	$[5, 10]$	1.8040	0.8716	1.3784	0.41	3.67
PINN + NDI	$[5, 10]$	1.6030	0.7125	1.1548	0.32	2.60
PINN + NDI + RBF	$[5, 10]$	1.3974	0.5936	0.8752	0.30	2.83

Step response metrics are evaluated separately for each command segment. The settling band is ±0.14°.

For the −20% case (Table A2), the overshoots following the command change at $t = 5$ s are 1.5647°, 1.1548°, and 0.9577° for PID, PINN + NDI, and PINN + NDI + RBF, respectively; the corresponding rise times are 0.41, 0.32, and 0.31 s, and the settling times are 3.91, 2.60, and 2.76 s. The RBF-compensated controller reduces the second-segment overshoot by 38.79% relative to PID. Over $[0, 10]$ s, its RMSE and MAE are 1.6321° and 0.6746°, which are 26.60% and 36.09% lower than the PID values. The absolute tracking error at $t = 10$ s is 0.0198°.

Table A2. Angle-of-attack tracking metrics under a −20% elevator effectiveness perturbation.

Strategy	Interval (s)	RMSE (°)	MAE (°)	$O_{\max}$ (°)	t_r (s)	t_s (s)
PID	$[0, 5)$	2.3982	1.0965	1.1828	0.49	2.60
PINN + NDI	$[0, 5)$	1.9090	0.7882	0.9148	0.35	2.84
PINN + NDI + RBF	$[0, 5)$	1.7873	0.7218	0.7907	0.33	2.94
PID	$[5, 10]$	2.0339	1.0147	1.5647	0.41	3.91
PINN + NDI	$[5, 10]$	1.6030	0.7125	1.1548	0.32	2.60
PINN + NDI + RBF	$[5, 10]$	1.4605	0.6274	0.9577	0.31	2.76

Step response metrics are evaluated separately for each command segment. The settling band is ±0.14°.

Across both perturbation cases, RBF compensation reduces overshoot and tracking error magnitude relative to PINN + NDI in both command segments. This improvement does not imply faster settling in every segment. In the first segment, the RBF settling times are 2.95 and 2.94 s for the +20% and −20% cases, respectively, which are longer than those of both comparison controllers. In the second segment, the RBF controller settles faster than PID but more slowly than PINN + NDI. All three controllers enter and remain within the specified settling band before the end of each command segment in the supplied records. These results quantify the tracking performance trade-off under the two prescribed parameter perturbations.

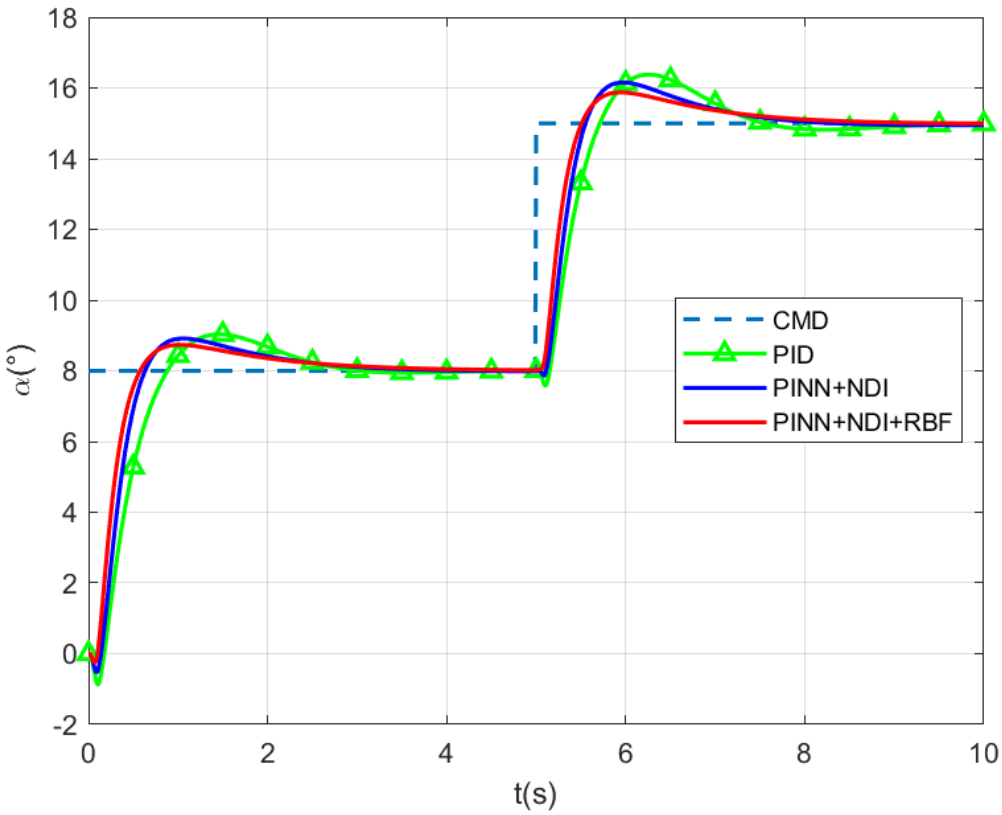


Figure A1. angle-of-attack tracking responses under a +20% elevator effectiveness perturbation.

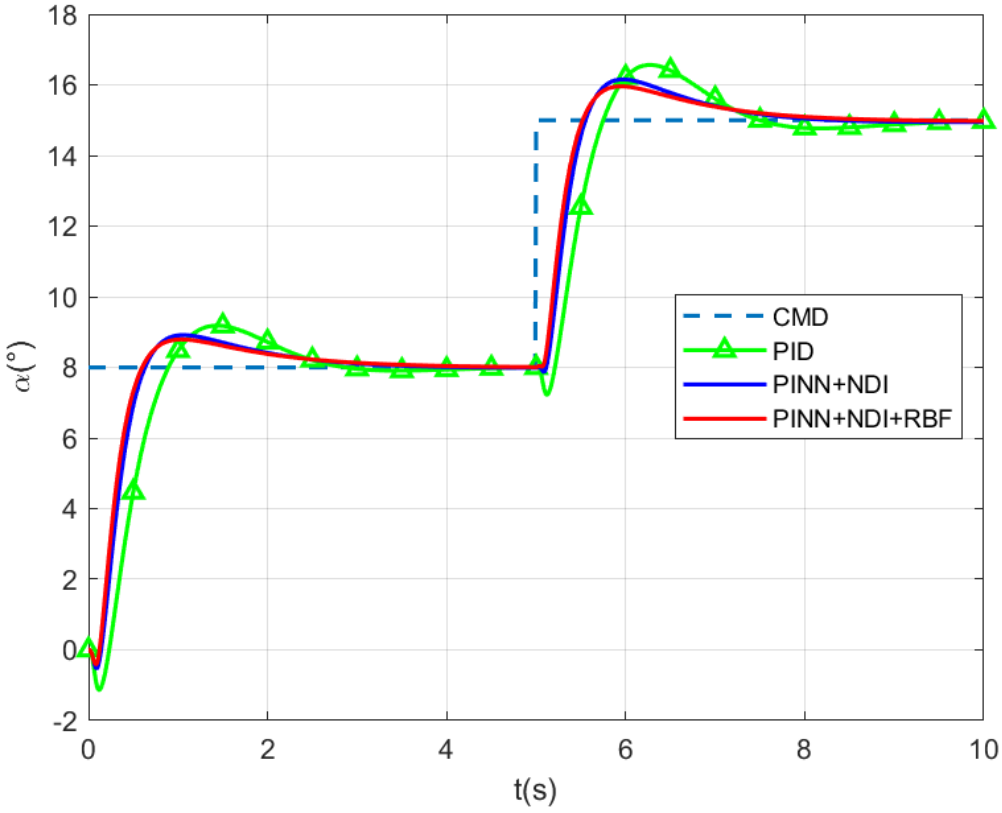


Figure A2. angle-of-attack tracking responses under a −20% elevator effectiveness perturbation.

References

1. Li, R.; Chen, G.; Lu, Y.; Qin, K.; Zhou, T.; Wang, W. Longitudinal Perching Trajectory Planning for a Fixed-Wing Unmanned Aerial Vehicle at High Angle of Attack Based on the Estimation of Region of Attraction. *Drones* **2025**, *9*, 87. [CrossRef]
2. Mi, B.; Cheng, S.; Zhan, H.; Yu, J.; Wang, Y. Development on Unsteady Aerodynamic Modeling Technology at High Angles of Attack. *Arch. Comput. Methods Eng.* **2024**, *31*, 4305–4357. [CrossRef]
3. Fukami, K.; Taira, K. Grasping Extreme Aerodynamics on a Low-Dimensional Manifold. *Nat. Commun.* **2023**, *14*, 6480. [CrossRef] [PubMed]
4. Stradtner, M.; Bekemeyer, P. Nonlinear Unsteady Aerodynamic Reduced-Order Modeling Using a Surrogate-Based Recurrent Framework. *J. Aircr.* **2025**, *62*, 1560–1574. [CrossRef]
5. Wang, X.; Shen, C.; Jiang, H.; Lin, Q.; Ji, Y. Optimal Design and Verifications of Cable-Suspension Mechanism for High-Angle-of-Attack Unsteady Tests. *J. Aircr.* **2023**, *60*, 2017–2023. [CrossRef]
6. Kim, S.; Lee, B.; Choi, S. Data-Driven Methods for High-Fidelity Aerodynamic Analysis of Controllability of Fighter Aircraft. In Proceedings of the AIAA AVIATION FORUM AND ASCEND 2024, Las Vegas, NV, USA, 29 July–2 August 2024. [CrossRef]
7. Le Clainche, S.; Ferrer, E.; Gibson, S.; Cross, E.; Parente, A.; Vinuesa, R. Improving Aircraft Performance Using Machine Learning: A Review. *Aerosp. Sci. Technol.* **2023**, *138*, 108354. [CrossRef]
8. Giernacki, W.; Stępień, S.; Chodnicki, M.; Wróblewska, A. Hybrid Quasi-Optimal PID-SDRE Quadrotor Control. *Energies* **2022**, *15*, 4312. [CrossRef]
9. Zhao, Z.-Y.; Tomizuka, M.; Isaka, S. Fuzzy Gain Scheduling of PID Controllers. *IEEE Trans. Syst. Man. Cybern.* **1993**, *23*, 1392–1398. [CrossRef]
10. Maaloul, B.; Elloumi, S. Adaptive PID Controller of a Quadrotor. In *Proceedings of the 2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6. [CrossRef]
11. Zhao, D.; Sun, C.; Wang, Q.; Yang, W. Neural Network Based PID Control for Quadrotor Aircraft. In *Intelligence Science and Big Data Engineering. Big Data and Machine Learning Techniques*; Springer: Cham, Switzerland, 2015; pp. 287–297. [CrossRef]
12. Taherinezhad, M.; Ramirez-Serrano, A. An Enhanced Incremental Nonlinear Dynamic Inversion Control Strategy for Advanced Unmanned Aircraft Systems. *Aerospace* **2023**, *10*, 843. [CrossRef]
13. de Heer, P.; de Visser, C.C.; Hoogendoorn, M.L.; Jentink, H.W. Incremental Nonlinear Control Allocation for an Aircraft with Distributed Electric Propulsion. In Proceedings of the AIAA SCITECH 2023 Forum, National Harbor, MD, USA, 23–27 January 2023. [CrossRef]
14. Tang, J. Research on a Thrust Vectoring Control Method for High Angle of Attack Based on Improved Incremental Nonlinear Dynamic Inversion. In *Proceedings of the 2025 11th International Conference on Mechanical Engineering, Materials and Automation Technology (MMEAT), Shenzhen, China*; IEEE: Piscataway, NJ, USA, 2025; pp. 35–45. [CrossRef]
15. Tian, Y.; Tan, Y.; Xu, B.; Liang, X. Dynamic Inversion-Based Flight Control for Supersonic Civil Aircraft with High Angle of Attack. In *Proceedings of the 6th China Aeronautical Science and Technology Conference*; Chinese Society of Aeronautics and Astronautics; Springer: Singapore, 2024; Volume III, pp. 718–726. [CrossRef]
16. Zhu, M.; Wang, J.; Zhang, X. Composite Control of Aerodynamic Direct Force and Thrust Vector for Carrier-Based Aircraft Based on Nonlinear Dynamic Inversion. In Proceedings of the 2025 44th Chinese Control Conference, Chongqing, China, 28–30 July 2025; pp. 4543–4548. [CrossRef]
17. Stougie, J.; Pollack, T.; van Kampen, E.-J. Incremental Nonlinear Dynamic Inversion Control with Flight Envelope Protection for the Flying-V. In Proceedings of the AIAA SCITECH 2024 Forum, Orlando, FL, USA, 8–12 January 2024. [CrossRef]
18. Stam, N.; de Visser, C.C. Adaptive Dynamic Incremental Nonlinear Control Allocation for Aircraft with Innovative Control Effectors. In Proceedings of the AIAA SCITECH 2025 Forum, Orlando, FL, USA, 6–10 January 2025. [CrossRef]
19. Li, Y.; Liu, X.; Ming, R.; Li, K.; Zhang, W. Improved Model Reference-Based Adaptive Nonlinear Dynamic Inversion for Fault-Tolerant Flight Control. *Int. J. Robust Nonlinear Control* **2023**, *33*, 10328–10359. [CrossRef]
20. de Visser, C.C.; Pool, D.M. Stalls and Splines: Current Trends in Flight Testing and Aerodynamic Model Identification. *J. Aircr.* **2023**, *60*, 1480–1502. [CrossRef]
21. Morelli, E.A.; Grauer, J.A. Practical Aspects of Frequency-Domain Approaches for Aircraft System Identification. *J. Aircr.* **2020**, *57*, 268–291. [CrossRef]
22. Hu, Y.; Yu, Z.; He, Y.; Zhang, H.; Jiang, L.; Yang, H.; Yu, Z. Aerodynamic Modeling and Aerodynamic Parameter Identification for High-Speed Aircraft. In Proceedings of the 2024 China Automation Congress, Qingdao, China, 1–3 November 2024; pp. 6202–6207. [CrossRef]
23. Lerro, A.; Trainelli, L.; de Pasquale, L. Identification of Longitudinal Aircraft Modes and Aerodynamic Parameters with Flight Trials. In Proceedings of the 2024 11th International Workshop on Metrology for AeroSpace (MetroAeroSpace), Lublin, Poland, 3–5 June 2024; pp. 24–29. [CrossRef]
24. Kumar, A.; Ghosh, A.K. Regularization Regression Methods for Aerodynamic Parameter Estimation from Flight Data. *Aircr. Eng. Aerosp. Technol.* **2023**, *95*, 820–830. [CrossRef]

25. Cao, R.; Lu, Y.; He, Z. System Identification Method Based on Interpretable Machine Learning for Unknown Aircraft Dynamics. *Aerosp. Sci. Technol.* **2022**, *126*, 107593. [CrossRef]
26. Dong, W.; Wang, X.; Lin, Q.; Cheng, C.; Zhu, L. A Weighted Feature Fusion Model for Unsteady Aerodynamic Modeling at High Angles of Attack. *Aerospace* **2024**, *11*, 339. [CrossRef]
27. Ma, J.; Zhu, Q.; Xue, T.; Ai, J.; Dong, Y. Aircraft Dynamics Modeling at High Angle of Attack Incorporating Residual Transformer Autoencoder and Physical Mechanisms. *Aerosp. Sci. Technol.* **2025**, *160*, 110045. [CrossRef]
28. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
29. Gu, W.; Primatesta, S.; Rizzo, A. Physics-Informed Neural Network for Quadrotor Dynamical Modeling. *Robot. Auton. Syst.* **2024**, *171*, 104569. [CrossRef]
30. Lin, J.; Chen, S.-S.; Yang, H.; Jiang, Q.-F.; Liu, J. A Physics-Informed Neural Network-Based Aerodynamic Parameter Identification Method for Aircraft. *Phys. Fluids* **2025**, *37*, 027200. [CrossRef]
31. Liu, C.; Song, W.; Xu, C.; Han, Z. Physics-Informed Data-Driven Nonlinear Unsteady Aerodynamic Modelling at High Angles of Attack. *Chin. J. Aeronaut.* **2026**, *39*, 103943. [CrossRef]
32. Xie, R.; He, H.; Meng, X.; Fan, X. Neural-Network-Based Dynamic Flight Envelope Protection Within Integrated Aircraft Control System. *J. Guid. Control Dyn.* **2023**, *46*, 1083–1094. [CrossRef]
33. Deng, B.; Xu, J.; Yuan, X.; Yu, S. Active Disturbance Rejection Flight Control and Simulation of Unmanned Quad Tilt Rotor eVTOL Based on Adaptive Neural Network. *Drones* **2024**, *8*, 560. [CrossRef]
34. Shen, Y.; Chen, M. Event-Triggering-Learning-Based ADP Control for Post-Stall Pitching Maneuver of Aircraft. *IEEE Trans. Cybern.* **2024**, *54*, 423–434. [CrossRef] [PubMed]
35. Liu, J.; Ji, Y.; Xu, L.; Chen, Z. Extended State Observer Based Generalized Predictive Control for Aircraft at High Angle of Attack. *Int. J. Control Autom. Syst.* **2024**, *22*, 2578–2590. [CrossRef]
36. Guerrero-Sanchez, M.E.; Montoya-Morales, J.R.; Valencia-Palomo, G.; Hernandez-Gonzalez, O. Robust IDA-PBC for non-separable PCH systems under time-varying external disturbances. *Nonlinear Dyn.* **2025**, *113*, 3499–3510. [CrossRef]
37. Xiong, Z.; Xu, Y.; Liu, D.; Pian, X. Robust cascaded control with antisaturation for fixed-wing trajectory tracking. *J. Aircr.* **2026**, *advance online publication*. [CrossRef]
38. Jia, K.; Lin, S.; Du, Y.; Zou, C.; Lu, M. Research on Route Tracking Controller of Quadrotor UAV Based on Fuzzy Logic and RBF Neural Network. *IEEE Access* **2023**, *11*, 111433–111447. [CrossRef]
39. Ruan, S.; An, S.; Dong, Z.; Jin, Z.; Liu, K. Integrated Control Method for STOVL UAV Based on RBF Neural Network and Nonlinear Dynamic Allocation. *Drones* **2025**, *9*, 167. [CrossRef]
40. Sukumar, N.; Srivastava, A. Exact Imposition of Boundary Conditions with Distance Functions in Physics-Informed Deep Neural Networks. *Comput. Methods Appl. Mech. Eng.* **2022**, *389*, 114333. [CrossRef]
41. Nabian, M.A.; Gladstone, R.J.; Meidani, H. Efficient Training of Physics-Informed Neural Networks via Importance Sampling. *Comput.-Aided Civ. Infrastruct. Eng.* **2021**, *36*, 962–977. [CrossRef]
42. Sanwale, J.; Singh, D.J. Aerodynamic Parameters Estimation Using Radial Basis Function Neural Partial Differentiation Method. *Def. Sci. J.* **2018**, *68*, 241–250. [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.