IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.
Cite as T-ASE paper.

Second-Order Sliding Mode Control of Flying-Wing Aircraft Based on Feedforward Neural Networks

Yuecheng Song, Zhenbao Liu, *Senior Member, IEEE*, Junwei Han, *Fellow, IEEE*, Jinbiao Yuan, Wen Zhao, and Qingqing Dang

***Abstract*—The flying-wing aircraft control problem is a major concern. In this paper, a new control strategy is introduced. First, a Feedforward neural network (FNN) modeling is introduced. Then, a second-order sliding mode control is applied, with the parameters generated from Deep Deterministic Policy Gradient (DDPG) reinforcement learning. To study the disturbance rejection performance, wind disturbance is applied to the aircraft using a deep neural network as an disturbance observer for different types of winds. Finally, All three simulations: Simulink, Software In The Loop, and Hardware In the Loop are applied to show the effectiveness of the proposed strategy. The simulation results show that the proposed method demonstrates good robustness in various conditions.**

***Note to Practitioners*—This paper is motivated by the traditional linearized flying-wing aircraft controller with the effectiveness of the FNN and reinforcement learning on UAV applications. The controller can be designed for various situations without changing the parameters by modeling the aircraft through the FNNs. The theoretical framework proposed in this paper combines the multiple-input multiple-output (MIMO) sliding mode strategy with the reinforcement learning with higher accuracy modeling. This method reduces the settling time, overshoot and steady error. More realistic effects should be considered for deployment into real aircraft. The parameters of the neural networks should also be adjusted in real applications.The FNN networks and DDPG have low computational footprints and are readily deployable on embedded systems like Pixhawk. The DNN observer may require model compression for the smallest processors. While the current framework requires per-aircraft training to achieve optimal performance, this process is conducted offline. The resulting controller gains are then fixed for reliable real-time operation, providing a clear pathway for implementation on specific UAV platforms.**

***Index Terms*—Flying wing, sliding mode, feedforward neural network, DDPG reinforcement learning, deep neural network.**

I. Introduction

With technology development, uncrewed aerial vehicle (UAV) applications have been applied to many fields. The novel flying-wing aircraft has many advantages, including high aerodynamic efficiency, perfect lift-drag characteristics, long flight endurance, and a simple structure. However, some disadvantages exist. Since the aircraft structure is the fusion of the wings and the body fuselage, there are no tail wings, which makes it difficult to stabilize.

Many researchers have addressed this problem. Many control strategies exist, such as [1] and [2]. In these methods, the control process is designed on a linearized model. These methods have good control at selected linearized points. However, if the current working point is different from the selected linearized point, the response will be worse because the flying-wing aircraft is a high variance nonlinearized model. Some control strategies have applied nonlinear models with good results. However, since nonlinear modeling of the flying-wing aircraft is too complex, this strategy has not been applied in many studies. The controller design is also a major challenge. Many control methods exist, such as proportional-integral-derivative (PID) [1], state feedback [2], H-infinity [3], linear quadratic [4], backstepping [5], integral backstepping [6], and sliding mode [7] etc. These methods can be separated into two groups. One group adds feedback to the actuators. By feeding the aircraft state back to the control input, the controller has a desired dynamic response, thus can be applied only to the linearized model. The other group feeds back the aircraft aerodynamic and replaces the original dynamics with the desired dynamic response. Since the controller is designed independently in these methods, they are robust regardless of modeling. Sliding mode control is a high-performance nonlinear control strategy. It has a finite convergence time and is popular in many UAV applications [8], [9]. It also has a good disturbance rejection ability. However, according to the authors in [10], chattering in the sliding mode control process damages the actuators. Moreover, fractional order

Received 18 June 2025; revised 29 August 2025; accepted 8 September 2025. Date of publication 23 September 2025; date of current version 7 October 2025. This article was recommended for publication by Associate Editor C. M. Abdissa and Editor C. Seatzu upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation Fund under Grant 52072309 and Grant 62303379, in part by the Key Program of the National Natural Science Foundation of China under Grant U2433208, in part by the Key Research and Development Program of Shaanxi under Program 2019ZDLGY14-02-01, in part by Shenzhen Fundamental Research Program under Grant JCYJ20190806152203506, in part by the Natural Science Foundation of Shaanxi Province under Grant 2023-JC-QN-0665, in part by Suzhou Municipal Science and Technology Bureau under Grant ZXL2023177, in part by the Aeronautical Science Foundation of China under Grant ASFC-2018ZC53026, in part by the Foundation of Yunnan Key Laboratory of Unmanned Autonomous Systems under Grant 202408ZD01, and in part by the National Key Laboratory Foundation of Helicoptor Aeromechanics under Grant 2024-ZSJ-LB-02-03. *(Corresponding author: Zhenbao Liu.)*

Yuecheng Song and Junwei Han are with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: illidan_st227@mail.nwpu.edu.cn; jhan@nwpu.edu.cn).

Zhenbao Liu, Wen Zhao, and Qingqing Dang are with the School of Civil Aviation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: liuzhenbao@nwpu.edu.cn; zhaowen@nwpu.edu.cn; dangqingqing@nwpu.edu.cn).

Jinbiao Yuan was with the School of Civil Aviation, Northwestern Polytechnical University, Xi'an 710072, China. He is now with Qing'an Group Company Ltd., Xi'an 710077, China (e-mail: yuanjinbiao0414@126.com).

Digital Object Identifier 10.1109/TASE.2025.3613383

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

sliding surface controller have also been played a key role in aircraft control [11], [12]. To solve this problem, some researchers have modified the structure of the sliding surface [13], [14], [15], [16]. Another strategy is to increased the order of the sliding surface, as in the work of [17], [18], and [19]. These studies show that chattering can be decreased by increasing the order of the sliding surface, such as super twisting controller [20], [21], [22], or terminal controller [23], [24].

Robust modeling is needed to stabilize flying-wing aircraft. Since the aircraft model is complex nonlinear, a simple linearized model is not suitable. The nonlinear model increases the robustness of the aircraft and is suitable for a large range, as found by many researchers [25], [26], [27]. However, the nonlinear model of flying-wing aircrafts is complex, and the statuses of the aircraft are closely coupled, making it difficult to apply a nonlinear control strategy. To overcome this, a more comprehensive method should be applied.

The BP(Backpropagation) network is one of the most popular neural networks. It is used for many control applications, such as BP network PID [28] and BP network disturbance measurement [29], [30]. In these methods, the BP network works as an estimation function with the parameters changing automatically since the form of the estimated part is unknown. In some applications, the BP network is applied for modeling an unknown model, as in [31]. In their research, the model parameters model did not change; thus, an offline BP network for modeling is suitable and sufficient. Although the nonlinear model is complex, it can be written in a closed form for flying-wing aircraft. Also, inspired by the BP network offlined modeling, the feedforward neural networks are also applied for function modeling, such as [32] and [33].

Another problem is aircraft controller design. Traditional control strategies have disadvantages in terms of parameter selection, especially for flying-wing aircraft. To avoid this, a new attitude control strategy should be introduced.

Wind disturbance is another problem for flying-wing aircraft. Solving this problem is important since wind disturbances can occur anywhere. Many studies have addressed this topic. These methods include state observers [34], [35], [36] or neural network observers [37], [38], [39]. In these methods, the wind disturbance is estimated through the observer and then fed back to the actuator to decrease the disturbance. The more precise the estimator is, the better the control performance will be. Traditional state observers include state feedback [35] or Kalman filters [40]. These methods introduce a higher convergence rate observer for disturbance estimation; however, they introduce noise to the controller since a large derivation is introduced. Besides, many researches have paid attention on the higher order disturbance observers such as [41], PDE based state observer [42] etc. They have introduced an higher order based disturbance observer, which automatically converge to the desired state. Also, adaptive based state observers are also included in control strategies such as [43] and [44]. In these research, the author using the adaptive control strategies to estimate and mitigate the disturbance from the aircraft. Neural network observers directly calculate the disturbance from the current status of the aircraft since the neural network [45], including the FNN network, RBF(Radical Basis Function) Network [46], RNN(Recurrent Neural Network) network [47] etc. can measure any function with a desired error. Neural networks can also update their own parameters to updated to the optimized parameters [48], [49], [50]. The more layers there are in the neural network, the more precise the modeling will be.

For aircraft control, the fuzzy controller has also played a key role in recent research. Fuzzy controller has the ability for choosing controller parameters automatically, and having a good robustness in applications, many researches have paid attention on it, such as [51], [52], and [53]. Besides, fuzzy control strategies often integrate with neural network controller such as ANFIS(Adaptive Network-based Fuzzy Inference System) [54], [55], which improves the robustness of the controller and learning the control parameters automatically.

Reinforcement learning is a new type of intelligent control strategy. It has been used in many applications, such as power [56], robotics attitude [57], and aircraft control [58]. Some researches have paid attention on the attitude control of the aircraft such as [59], [60], and [61]. In these papers, the controller performs well and achieves good robustness. Some researches have paid attention to safe deep reinforcement learning control strategy [62], [63], and the aircraft in the framework can automatically avoid the collisions with a good performance. However, for flying wing aircraft, there are too few available algorithms for attitude control, making the control process challenging. In this paper, a deep reinforcement learning control strategy is introduced to the attitude control process.

A deep network is a neural network with more layers that can extract deep features of normal signals. The deep neural network has been applied in pattern recognition [64], [65], signal processing [66], and aircraft control [67], [68]. Since deep neural networks are more powerful tools than traditional neural networks, deep neural networks have become a new aircraft control application.

The mass variation control is also a major concern. For fuel-based aircraft, the aircraft mass decreases every second. In artificial rain applications, aircraft mass also decreases continuously. In these applications, the mass decreases in seconds. If weight variations are neglected in the modeling process, the aircraft's performance will be significantly impacted. To overcome this, the weight of the aircraft should be modeled in the control system. However, in most research, aircraft mass is considered independently, as in [69].

To address the core challenges associated with the control of flying-wing aircraft, this paper proposes a novel, integrated control strategy. The approach employs a FNN for high-fidelity aircraft modeling, a second-order sliding mode controller for robust attitude stabilization, and a DNN-based observer for accurate wind disturbance estimation. Furthermore, the framework explicitly incorporates the aircraft's time-varying mass to ensure adaptability across different operational conditions. The major contributions of this work are summarized as follows:

1. A Novel FNN-Based Modeling Framework for Enhanced Generality and Accuracy: This paper introduces a systematic offline modeling strategy

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

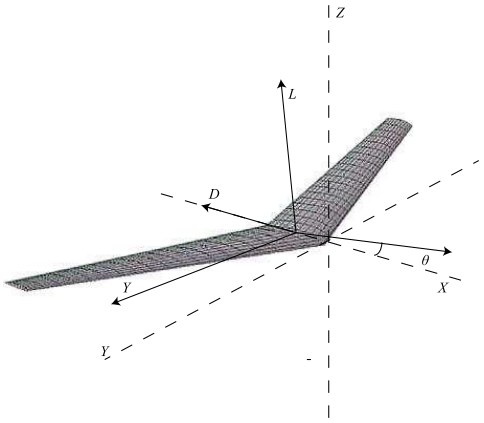

Fig. 1. The UAV for force analysis.

using FNNs to create a unified nonlinear model of the aircraft dynamics. This model captures complex aerodynamic dependencies across the entire flight envelope, significantly improving modeling accuracy and generalization capability compared to traditional methods limited to specific operating points.

2. An Intelligent, Self-Tuning Control Architecture via Deep Reinforcement Learning: Building upon the FNN model, we develop an auto-tuning control mechanism based on the DDPG algorithm. This innovative contribution leverages deep reinforcement learning to dynamically optimize the parameters of a MIMO second-order sliding mode controller, ensuring robust and near-optimal performance without manual gain scheduling.
3. A Unified Robust Framework with Deep Learning-Based Disturbance Observer: To fortify the control system against external disturbances, a dedicated DNN-based observer is seamlessly integrated into the proposed strategy. This component provides accurate real-time estimates of wind disturbances, enabling proactive compensation and validating the framework's robustness and practical effectiveness in challenging windy environments.

The force analysis is introduced in section II, and the attitude control is introduced in section III. The robustness analysis is performed in section IV. The simulation is applied in section V, and we conclude in section VI.

II. Force Analysis

The force analysis is illustrated in Fig. 1. In the framework, the coordinate axis is constructed on the aircraft body frame, with the heading forward X-axis, right side Y-axis, and downside of the aircraft Z-axis. The body axes are denoted as x_b, y_b, and z_b. For the Earth coordinates, the axes can be illustrated as x_e, y_e and H. As illustrated in [70] and [71], the aircraft is affected by drag, thrust, lift, side force and gravity.

Additionally, flying-wing aircraft are affected by moments on the aircraft. According to [71], all the drag, lift and side force are proportional to the wing surface of the aircraft and the square of the current speed, and the forms can be written as $L = \frac{1}{2}C_L\rho S V^2$, $D = \frac{1}{2}C_D\rho S V^2$, and $Y = \frac{1}{2}C_Y\rho S V^2$. In the equation, the terms C_L, C_D, and C_Y are the lift, drag and side force coefficients, respectively. C_D and C_L are related to the current speed V and the angle of attack α. The derivative equation of the aircraft can be written as follows:

$$\begin{cases} \dot{V} = \frac{T\cos\alpha\cos\beta - D - mg\sin(\theta-\alpha)}{m} \\ \dot{\alpha} = \frac{-T\sin\alpha - L}{mV\cos\beta} + \\ \frac{V(-p\cos\alpha\sin\beta + q\cos\beta - r\sin\alpha\sin\beta) + g\cos(\theta-\alpha)}{V\cos\beta} \\ \dot{\beta} = \frac{-T\cos\alpha\sin\beta + Y}{mV} - (-p\sin\alpha + r\cos\alpha) \\ \dot{p} = \frac{((I_{yy}I_{zz} - I_{zz}^2 - I_{xz}^2)r + (I_{xx}I_{xz} - I_{yy}I_{zz} - I_{xz}I_{zz})p)q + I_{zz}L_A + I_{xz}N_A}{I_{xx}I_{zz} - I_{xz}^2} \\ \dot{q} = \frac{(I_{xx} - I_{zz})pr - I_{xz}(p^2 - r^2) + M_A}{I_{yy}} \\ \dot{r} = \frac{((I_{xx}^2 - I_{xx}I_{yy} + I_{xz}^2)p - (I_{xx}I_{xz} - I_{yy}I_{zz} - I_{xz}I_{zz})r)q + I_{xz}L_A + I_{xx}N_A}{I_{xx}I_{zz} - I_{xz}^2} \\ \dot{\phi} = p + (r\cos\phi + q\sin\phi)\tan\theta \\ \dot{\theta} = q\cos\phi - r\sin\phi \\ \dot{\psi} = \frac{r\cos\phi + q\sin\phi}{\cos\theta} \\ \dot{x} = V\cos(\theta-\alpha)\cos(\psi+\beta) \\ \dot{y} = V\cos(\theta-\alpha)\sin(\psi+\beta) \\ \dot{H} = V\sin(\theta-\alpha) \end{cases} \tag{1}$$

In the equation above, the terms I_{xx}, I_{yy}, I_{zz}, and I_{xz} are the moments of inertia on each axis. The terms L_A, M_A, and N_A are the moments on the aircraft, which can be expressed as follows:

$$\begin{cases} L_A = \frac{1}{2}\rho V^2 S_w b\left(C_{l\beta}\beta + C_{l\delta_a}\delta_a + Cl_{\delta_r}\delta_r + \right. \\ \left. C_{lp}\frac{bp}{2V} + C_{lr}\frac{br}{2V}\right) \\ M_A = \frac{1}{2}\rho V^2 S_w c_A\left(Cm + C_{m\alpha}\alpha + C_{mq}\frac{qc_A}{2V} + \right. \\ \left. C_{m\delta_e}\delta_e + C_{\delta_T}\delta_T\right) \\ N_A = \frac{1}{2}\rho V^2 S_w b\left(C_{n\beta}\beta + C_{n\delta_a}\delta_a + \right. \\ \left. Cn_{\delta_r}\delta_r + C_{np}\frac{bp}{2V} + C_{nr}\frac{br}{2V}\right) \end{cases} \tag{2}$$

In the equation, the terms $C_{l\beta}$, $C_{m\alpha}$ are the coefficients of the flying wing of the aircraft based on the current flying status. The parameters δ_e, δ_a, and δ_r are the actuator inputs, and the term δ_T is the engine thrust.

III. Attitude Control

The control strategy can be designed with the flying-wing aircraft parameters defined above as follows. The derivative equation of the aircraft can be shown as follows:

$$\dot{x} = f(x) + g(x)u \tag{3}$$

In the equation above, the term $f(x)$ is the aircraft state function, and $g(x)$ is the input coefficient function. The input vector includes $[V, \alpha, \beta, p, q, r]$, and the input u includes δ_e, δ_a, δ_r, and δ_T.

$$\dot{x} = f(V, \alpha, \beta, p, q, r) + g(\delta_e, \delta_a, \delta_r, \delta_T) \tag{4}$$

The nonlinear form of the aircraft is illustrated in (4). The term in the speed and angle-of-arrival (AoA) channel makes it very difficult to separate the variables. To avoid this, a FNN network is applied for modeling. Through Taylor expansion,

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

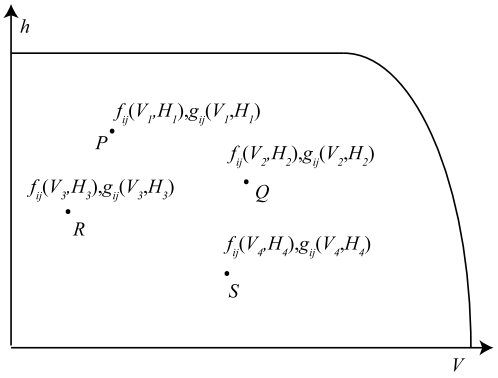

Fig. 2. The dimensional derivative functions at different point in the flight envelope.

the derivative equation of the aircraft can be written in this form:

$$\dot{x}_i = \sum_{j=1}^{6} f_{ij}(x) + \sum_{j=1}^{4} g_{ij}(x)u \tag{5}$$

In the equation above, the terms $f_{ij}(x)$ and $g_{ij}(x)$ are the aircraft modeling functions, and x_i denotes the aircraft state. At a specific linearization point, these functions become dimensional derivatives. All available points in the flying envelope are illustrated in Fig. 2.

In Fig. 2, the functions $f_{ij}(x)$ and $g_{ij}(x)$ at point P and point Q are different. According to (1), the derivative equation has a closed form. Thus, an FNN network is introduced for modeling the functions, and the derivative equations of the aircraft can be illustrated as follows:

$$\dot{x} = A(x)x + B(x)u \tag{6}$$

In the equation above, the functions $A(x)$ and $B(x)$ are the modeling functions calculated through the FNN network. This equation is the combination of the matrix form of (5). There are 60 variables in the function, in which $A(x)$ is 36 and $B(x)$ is 24. The function can be simplified since for the coupled parts between the longitudinal and lateral channels, the absolute value of these functions reaches 10^{-10}, while that of the remaining parts is no less than 10^{-3}. Since these terms are too small, and smaller than the absolute error itself, if they were considered into the framework, the error itself of the dimensional derivatives would be too large. Thus, by neglecting these parts, the control channels can be decoupled into longitudinal and lateral channels.

To stabilize the flying-wing aircraft, the following sliding surface is chosen:

$$S = x - x_d + (A_0 - B_0K)\int_0^t (x - x_d)dt \tag{7}$$

In the equation above, the matrices A_0 and B_0 are the state matrices from a randomly chosen point in the flying envelope from $A(x)$ and $B(x)$, respectively, and K is the state feedback matrix to ensure that the matrix $A_0 - B_0K$ is Hurwitz. This matrix does not change even though the matrix $A(x)$ and $B(x)$ change. The vector x is the current state of the aircraft; in the longitudinal channel, it includes the speed V, the angle of attack α, and the pitch angular rate q. In the lateral channel, the vector x includes the side slip angle β, the roll rate p, and the yaw rate r. From the nonlinear model applied in (1), the pitch rate q should also be included. The term x_d is the desired state.

To stabilize the angular rates of the aircraft, the sliding mode function is chosen as follows:

$$\dot{S} = -P_1S - Q_1 sgn(S) \tag{8}$$

In the equation above, the matrices P_1 and Q_1 denote the parameters of the sliding surface, both of which are positive definite.

The equation in (8) is the first-order sliding function, as introduced by many authors. However, chatter in the control process is harmful to the controller. To overcome this, a second-order sliding surface is introduced, shown as follows:

$$\dot{S} = -P_1S - Q_1 sgn(S) + e_1 \tag{9}$$

In (9), the term e_1 denotes the addition of a second-order sliding surface. The derivation is given below:

$$\dot{e}_1 = -R_1 S^{\frac{2a+1}{2b+1}} \tag{10}$$

In the equation above, the matrix R_1 denotes the matrix coefficient, while the parameters $2a + 1$ and $2b + 1$ are constant parameters, making the integral part an odd function, as illustrated in [72]. To prove the stability of the system, consider the following Lyapunov equation

$$V = \frac{1}{2\left(1 + \frac{2a+1}{2b+1}\right)} \sum S_i^{1+\frac{2a+1}{2b+1}} + \frac{R_1}{2}\left(\int_0^t S^{\frac{2a+1}{2b+1}}dt\right)^T \left(\int_0^t S^{\frac{2a+1}{2b+1}}dt\right) \tag{11}$$

The derivative of the Lyapunov equation can be written as follows:

$$\dot{V} = -P_1 \|S\|^{1+\frac{2a+1}{2b+1}} - Q_1 \|S\|^{\frac{2a+1}{2b+1}} \tag{12}$$

From the equation above, it can be found that the derivative of the Lyapunov equation is stable. This means that the angular rate channel is stable. Since the attitude angles are affected by the angular rates directly, and angular rates converge to steady as mentioned above, and the matrix A_c is bounded. The attitude angle control of the proposed method is illustrated as follows:

$$\omega_d = A_c^{-1}\dot{\Omega}_d + A_c^{-1}K_\Omega(\Omega - \Omega_d) \tag{13}$$

In the equation above, the term ω_d is the desired angular rate of the aircraft. The matrix K_Ω is the proportional matrix of the outer loop control; it is positive definite. The term Ω is the current attitude angle of the aircraft, and the term Ω_d is the desired attitude. The matrix A_c is the matrix form of the 7th to 9th equations in equation (1). If there were no second order sliding mode term, then the stability of the can be provided through a similar path to [73]. Due to the limitations that there is no term greater than 2nd order in the sliding surface, the stability proof seems not applicable. However, because the Lyapunov equation of the angular rates and the sliding surface design itself, the angular rates are stable. Combining

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

the derivative equation above and (1), the following derivative equation holds:

$$\dot{\Omega} - \dot{\Omega}_d = -K_\Omega(\Omega - \Omega_d) + A_c(\omega - \omega_d) \tag{14}$$

In the equation above, it can be found clearly that the attitude angular rates converge in an exponential speed, since the matrix K_Ω is positive definite, the attitude is stable.

A. Reinforcement Learning

As mentioned above, the parameters of the controller are chosen by hand. However, the proposed method has a disadvantage because of oscillations in the airframe even when the parameters are chosen correctly. This is because the flying-wing aircraft is a highly nonlinear model, even though the aircraft itself is modeled through the FNN network. To overcome this, a reinforcement learning control strategy is introduced.

The deep deterministic policy gradient (DDPG) agent is a powerful deep reinforcement learning agent. It has the following advantages. First, in any framework, it can calculate the desired parameters automatically. Additionally, it has strong disturbance rejection performance and thus can be applied in aircraft control.

The DDPG agent can be considered to have the following function:

$$L(\theta^Q) = \frac{1}{N}\sum_{i=1}^{N}(y_i - Q(s_i, a_i|\theta^Q))^2 \tag{15}$$

In the equation above, s_i denotes the i-th state. The function $L(\theta^Q)$ denotes the optimization function of DDPG network, and a_i is the action chosen by deep reinforcement learning, where θ^Q is the parameters of the critic network. This function optimizing the critic network, that evaluates the performance of the current actor. For the actor network, the training function is shown as follows:

$$J(\theta^\mu) = E^\mu\left[R_t|s_t, a_t\right] \tag{16}$$

In the equation above, term $J(\theta^\mu)$ is the actor optimization function. θ^μ is the parameters of the actor network. R_t is the reward function of current episode t. For the network optimization, the function is shown as follows [74]:

$$\begin{cases} \nabla_{\theta^Q} L(\theta^Q) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(y_i - Q(s_i, a_i|\theta^Q))\nabla_{\theta^Q} Q(s_i, a_i) \\ \nabla_{\theta^\mu} J(\theta^\mu) \approx \dfrac{1}{N}\displaystyle\sum_{i}\nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)}\nabla_{\theta^\mu}\mu(s|\theta^\mu)|s_i \\ \theta^Q_{t+1} = \theta^Q_t + \beta_Q\nabla_{\theta^Q} L(\theta^Q) \\ \theta^\mu_{t+1} = \theta^\mu_t + \alpha_\mu\nabla_{\theta^\mu} J(\theta^\mu) \end{cases} \tag{17}$$

In the equation above, the parameters β_Q and α_μ are the training parameters of critic network and actor network.

B. FNN Network

For the FNN network, the following sigmoid function is applied:

$$y = \frac{2}{1 + \exp(-(w^T x + b))} - 1 \tag{18}$$

In the equation, w is the weight of the current layer, and b is the bias of the neural network. x is the state of the FNN network at the current layer, and y is the output. The training function of the network can be given as follows

$$f = \frac{1}{2}r^T r = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}(y_{ij} - y_{ij0})^2 \tag{19}$$

In the equation above, r is the error between the neural network result and the true dimensional derivatives, and M and N denote the number of training samples and outputs of the FNN network, respectively. The training method can be applied with Levenberg-Marquardt algorithm. The update strategy is shown as follows:

$$x_{new} = x_{old} - (J^T J + \mu I)J^T r \tag{20}$$

In the equation above, J denotes the Jacobian matrix of r. $J^T r$ is the gradient of the function f, and $J^T J$ is the quasi Hessian matrix. The parameter μ is the damping factor in the optimization, which is updated automatically through the training process. x_{old} is the current training variable step, and x_{new} is referred to as the next step.

IV. Controller Robustness

In general, flying-wing aircraft are vulnerable to disturbances. Additionally, since the aircraft is modeled based on a FNN network, the model effectiveness should also be considered. To solve this problem, a deep learning method is applied.

A. Wind Disturbance

The first aircraft disturbance is wind. The flying-wing aircraft can be affected by wind disturbance due to its unstable structure. Traditional methods, such as a state observer or a neural network observer, use modeling strategies for wind disturbance estimation and have achieved good results in simulations. In the proposed method, since the aircraft is modeled through the FNN network, the derivative equation of the wind effect can be illustrated as follows:

$$\dot{x} = A(x)x + B(x)u + f_w \tag{21}$$

In the equation above, the term f_w denotes the wind effect. Since the wind affected parts include the current status, the functions $A(x)$ and $B(x)$ are affected. The equalized derivative equation can be expressed as follows:

$$\dot{x} = (A(x) + \Delta A)x + (B(x) + \Delta B)u \tag{22}$$

In the equation above, the matrices ΔA and ΔB denote the modeling errors between the artificial model and the constructed model in windy environments. Both satisfy

$$\Delta(A, B) \le \varepsilon_{FNN} \tag{23}$$

The term ε_{FNN} is the optimized root mean squared error (RMSE) of the training process in (19). According to this, the derivative equation of (22) can be written as follows:

$$\begin{cases} \dot{S} = (A(x) + \Delta A)x - (A_0 - B_0 K)x + \\ (B(x)^T B(x))^{-1}(B(x) + \Delta B)B(x)^T h(x) \end{cases} \tag{24}$$

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

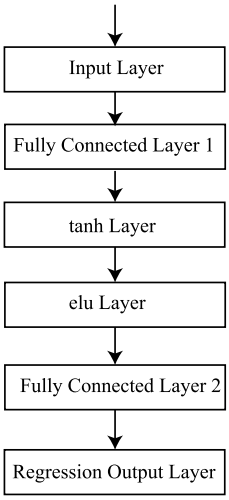

Fig. 3. The structure of the deep neural network.

In the equation above, the term $-F(S)$ denotes the right side of equation (9). By combining the term of ΔA and ΔB together, the wind effect of the aircraft f_w can be illustrated as follows:

$$f_w = \Delta A x + \Delta B(B(x)^T B(x))^{-1} B(x) h(x) \tag{25}$$

where

$$h(x) = -F(S) + (A_0 - B_0 K)x - A(x)x \tag{26}$$

Since the state of the aircraft is bounded and the optimized RMSE is small enough, the wind effect satisfies

$$f_w \leq L \tag{27}$$

In the equation above, the term L can be considered a constant. The Lyapunov equation satisfies

$$\dot{V} \leq -P_1 \|S\|^{1+\frac{2a+1}{2b+1}} - (Q_1 - L)\|S\|^{\frac{2a+1}{2b+1}} \tag{28}$$

According to the previous section, the attitude of the flying-wing aircraft is stable if and only if

$$\min \lambda_{Q_1} \geq L \tag{29}$$

where λ_{Q_1} denotes the eigenvalues of matrix Q_1. The general process is TECS (total energy control system) for height control, which is a combination control strategy based on both pitch angle and airspeed [75]. In this paper, since the airspeed is stabilized previously, the TECS height control can be transferred to a traditional PID. Thus, the error affected by the wind can be mitigated automatically.

For lateral position control, the control process can be considered waypoint-based control. Flying at three different way points A, B and C can be equivalent to achieving $180 - \angle ABC$ degree yaw control at point B. Since there are integrals in attitude control, as long as the attitude control process is stable, the aircraft can track the desired path.

B. Deep Neural Network

To decrease the wind disturbance, a neural network-based measuring strategy is introduced. Neural networks can fit any function with the desired precision. Online models can also have self-learning abilities. The more layers the network has, the higher the measurement precision will be. In most applications, the neural network has only 3 layers, such as the radical basis function (RBF) or recurrent neural network (RNN). In some applications, neural networks with more layers have also been applied [76]. According to [77], a deep neural network performs better than traditional neural networks. Thus, in this paper, a deep neural network is applied for wind disturbance measurement.

The deep neural network applied for wind disturbance has 6 layers. The first layer is the input layer. The second layer is a fully connected layer. The third layer is a traditional hyperbolic tangent layer. The fourth layer is an eLU function layer, which has a faster training time than the traditional sigmoid function or hyperbolic tangent function. The fifth layer is another fully connected layer, and the sixth layer is a regression layer. The structure of the deep neural network is shown in Fig. 3; a diagram of the proposed method is shown in Fig. 4. The reason is that the tanh function has the feature decentralization and smoother, making the inference faster, while the elu layer cannot be saturated in training, and having a wider range.

In Fig. 4, the terms $f(x)$ and $g(x)$ are nonlinear models of the function; they can be written as $A(x)x$ and $B(x)$, respectively, as illustrated in the previous sections. Since the number of inputs is not the same as the number of states, a pseudoinverse is applied. The arrows on the DNN (deep neural network) indicate that the deep neural network is tuned online, and no arrows on the FNN network indicate that the parameters of the FNN network do not change after training is completed. The terms $G_{act}(s)$ and $G_{wind}(s)$ denote the dynamics of actuators and wind disturbance, respectively. The input signal x_d denotes the desired attitude input, while the output signal x is the real attitude of the aircraft.

C. Parameter Errors

In real-time applications, the state variables of the aircraft are estimated through the external Kalman filter (EKF). In this method, the state of the aircraft is fused with the GPS and inertial navigation system (INS), and noise occurs in the fusion process. This leads to errors in the estimated states. In traditional control strategies, it only affects the feedback on each channel. However, in the proposed method, the aircraft modeling network is based on the current attitude. Parameter errors should be considered to ensure modeling effectiveness.

Considering the model with modeling errors, the derivative equations of the aircraft can be illustrated as follows:

$$\dot{V} = A(x + \Delta x)(x + \Delta x) + B(x + \Delta x)u \tag{30}$$

In the equation above, the term Δx is the parameter error. $A(x+\Delta x)$ and $B(x+\Delta x)$ are the modeling functions with input errors. The derivative equation can be illustrated as follows:

$$\dot{V} = A(x)x + B(x)u + A(x)\Delta x + \Delta A(x + \Delta x) + \Delta B u \tag{31}$$

In the equation, ΔA and ΔB are the modeling errors between the models with and without parameter errors. Similar to the previous proof, the aircraft is stable if the following equation is satisfied:

$$|A(x)\Delta x + \Delta A(x + \Delta x) + \Delta B u| \leq \min \lambda_{Q_1} \tag{32}$$

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

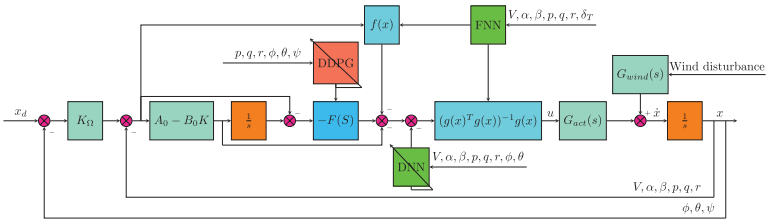

Fig. 4. The control process of the proposed method, $-F(S)$ denotes the sliding surface applied in Equation (9).

TABLE I
PARAMETER OF THE FLYING WING AIRCRAFT IN THE PAPER

Wingspan	Chores	MAC	Aspect Ratio	Surface
$2b$	c_0	c_A	λ	Surface
2000mm	350mm	263.33mm	8.00	$0.50m^2$

TABLE II
MASS AND MOMENTS INERTIAS OF THE AIRCRAFT

Parameters	Values
I_{xx}	$1.051 \times 10^5 kgmm^2$
I_{yy}	$1.255 \times 10^4 kgmm^2$
I_{zz}	$1.176 \times 10^5 kgmm^2$
I_{xz}	$255.5\ kgmm^2$
m	0.50kg

TABLE III
AERODYNAMIC PROPERTIES OF THE AIRCRAFT

Parameters	Values	Parameters	Values
C_{DV}	0.11245	$C_{D\alpha}$	-0.81166
C_{LV}	0.0032135	$C_{L\alpha}$	3.9934
C_{Lq}	3.1851	C_{mV}	0.021324
$C_{m\alpha}$	0.071848	C_{mq}	-2.4487
$C_{Y\beta}$	-0.0024922	C_{Yp}	0.26198
C_{Yr}	-0.06725	$C_{l\beta}$	-0.16039
C_{lp}	-0.45055	C_{lr}	0.31074
$C_{n\beta}$	0.03903	C_{np}	-0.18904
C_{nr}	0.0028225	$C_{L\delta e}$	0.3980
$C_{m\delta e}$	-1.9210	$C_{Y\delta a}$	-0.0296
$C_{l\delta a}$	-0.0992	$C_{n\delta a}$	-0.0039
$C_{Y\delta r}$	0.1158	$C_{l\delta r}$	0.0069
$C_{n\delta r}$	-0.0827	$\frac{\partial T}{\partial \delta_T}$	0.4436
$\frac{\partial L}{\partial \delta_T}$	0.0044	$\frac{\partial M}{\partial \delta_T}$	0.0012

V. SIMULATION RESULTS

A. Simulink Simulation

In this section, the simulation results are given and analyzed. The simulation is based on aircraft model introduced in [70]. The parameters of the aircraft are illustrated in Table. I and II. The aerodynamic derivatives are shown in Table. III. The platform is on a Ryzen 7 CPU with MATLAB R2024a. The simulation is based on the nonlinear model constructed through the flight dynamic toolbox [71]. The aircraft parameters are given as follows:

For the attitude control strategy, the controller is designed on both the longitudinal and lateral sides. For the longitudinal channel, the variables include the speed V, the AoA α, and the pitch rate q, and the inputs of the actuators include the engine δT and the pitch actuator δe. The matrix K is calculated through the Reccati equation, with the matrices A_0 and B_0 selected at 16 m/s with an engine input of 0.25. The matrices Q and R are selected as follows: $Q = diag\,[27, 16, 9]$ and R with only eigenvalues 100. For the lateral channel, the parameters are the side slip angle β, the roll rate p, the pitch rate q and the yaw rate r. The method is similar to that used for the longitudinal channel. The matrix Q is selected as follows: $Q = diag\,[16, 9, 9, 9]$, and the matrix R is another diagonal matrix with 100 eigenvalues.

The inputs for the FNN network training are the current attitude and the engine input of the aircraft; the outputs are the dimensional derivatives at different flying status. These parameters are generated through the following process: first, calculating all these parameters at all available steady flying points of the aircraft, then using the aircraft parameters as the input, these calculated data as the output for the training process. The original range of each parameter is shown as follows: airspeed V between 8m/s to 20m/s, AoA α and sideslip β between $-\pi/9$ and $\pi/9$, and angular rates of roll, pitch, and yaw between $-\pi/6$ and $\pi/6$. Then both inputs and outputs are normalized between -1 and 1. Because every dimensional derivative has a different form, these parameters can be trained separately. The FNN network is constructed with 3 layers, and the hidden layer has 80 nodes. The training method is based on Levenberg-Marquardt strategy [78], with the initial damping factor μ set to the maximum absolute eigenvalue of the matrix J^TJ. For each FNN network, the initial parameters are chosen randomly, and there are three stopping conditions. First, the change in the norm of the training parameter x is less than a threshold ε_1 times the variable norm. The second condition is that the gradient norm J^Tr is smaller than a threshold ε_2. The third condition is a failsafe, if any of the training variables reach infinity or not a number, these thresholds are both set to 10^{-4}. No training time limitation is applied to decrease the training errors as much as possible,. After 24 hours, the training process is complete. For each dimensional derivative, the function f reaches 10^{-9}, and the RMSE is no greater than 10^{-6}.

The following parameters are chosen for the aircraft attitude: the matrix P_1 has an eigenvalue of 4, and the matrix Q_1 has an eigenvalue of 1/15. For the second-order sliding mode part,

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

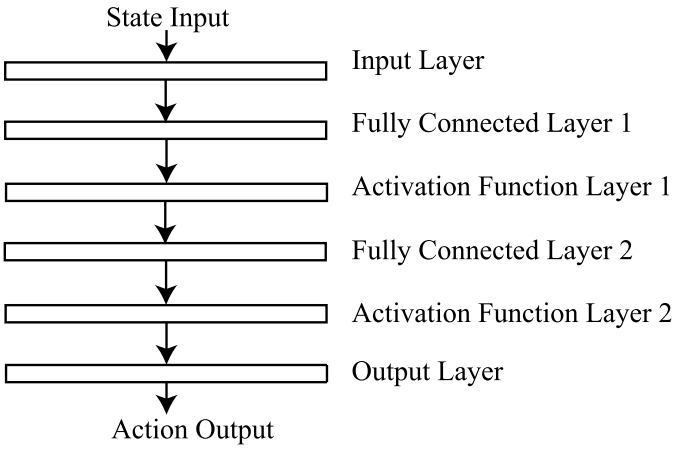

Fig. 5. The structure of the Actor Network.

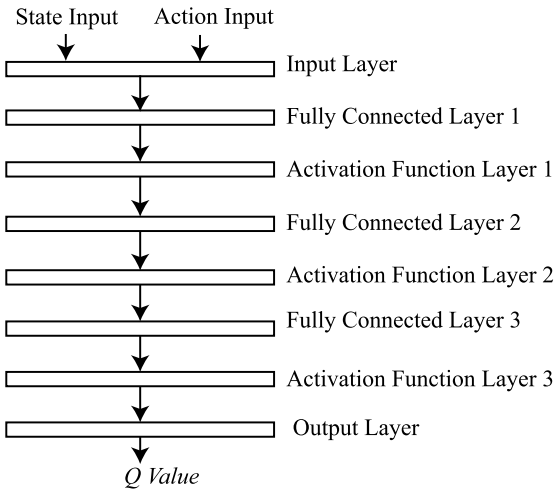

Fig. 6. The structure of the Critic Network.

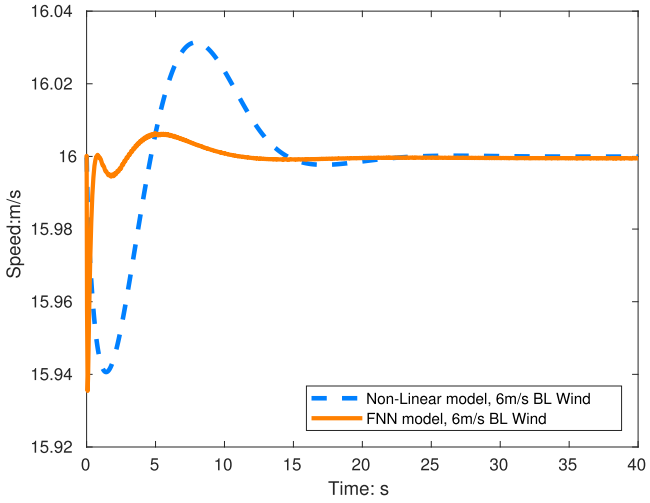

Fig. 7. Comparison of the speeds of the nonlinear and FNN network models with wind at 16m/s airspeed.

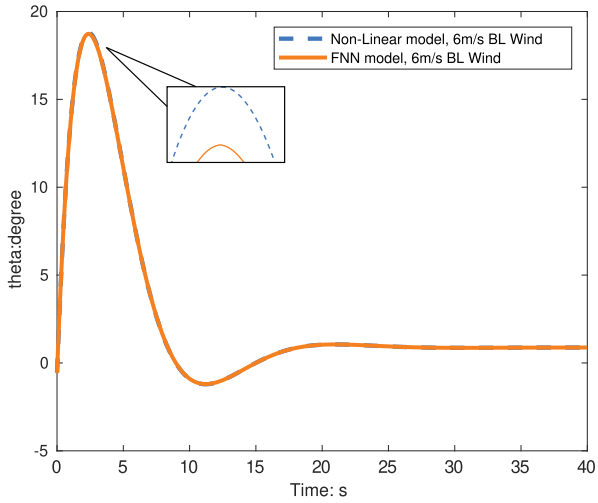

Fig. 8. Comparison of the pitch angles of the nonlinear and FNN Network models with wind at 16m/s airspeed.

the eigenvalue of R_1 is 1/15, with the parameter a set to 2 and b set to 5. For the attitude outer loop, the matrix K_Ω is set to 1/2. For the height control of the aircraft, a proportional integral (PI) control-based TECS control strategy is applied. Since the airspeed was previously stabilized in the airspeed loop, the remaining part is the height control directly on the pitch angle. In the outer loop, the parameter K_h is set to 1/2. For the inner loop, K_{Ih} is set to 1/3. The comparison between the nonlinear model and the FNN based model is shown in Fig.7, Fig. 8

The reinforcement learning parameters are as follows: both the actor network and critic network are 3 layers, with the second layer as the hyperbolic tangent function and the third layer as the fully connected layer. The desired reward function of the reinforcement learning can be shown as follows:

$$R(x) = \exp(-a_1 \|\Delta\phi\Delta\theta\Delta\psi\|) + \sum_{i=1}^{3} \tanh\left(\left\|\omega_i\Omega_i - \frac{k_i}{\omega_i\Omega_i}\right\|\right) \tag{33}$$

By optimizing the DDPG reward function $R(x)$, the reinforcement learning agent is applied to attitude control of the aircraft.The reason is that in the DDPG training, when the all attitude reach near the small range of the steady point, it can be considered that the aircraft has entered the steady area, thus the smaller the multiplication is, the larger the reward should be. The second term is the multiplication of the current attitude and the corresponding angular rates. When this multiply is small enough, this channel can be considered stable. The function makes sure that the smaller the error is, the higher the reward should be. The tanh function makes the unbounded function in to boundary, and the smaller the error is, the larger the reward will be. The structure of the actor network is shown in Fig. 5, and the structure of the critic network is shown in Fig. 6. For each hidden layers, there are 128 nodes for both actor network and critic network. With size 10^{-3} noise integrated to the training process. The training is based on the attitude control of the sliding mode itself, based on the converge speed parameter P_1, and the converge parameter K_Ω in (13).The reason of choosing the following sliding mode function is that the structure of the flying wing aircraft is complex, and designing the controller is difficult,

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

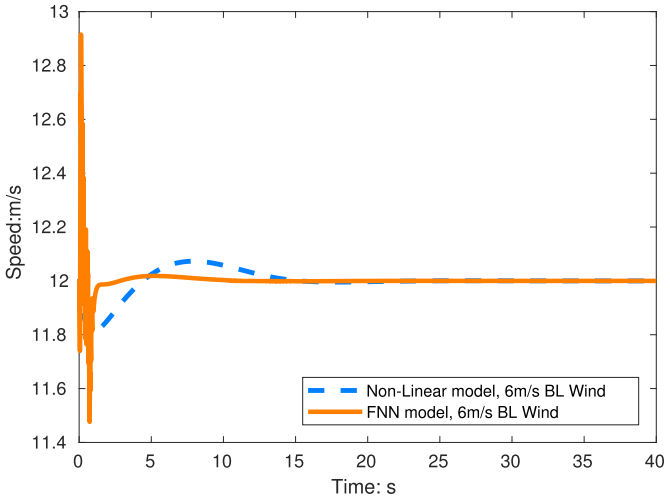

Fig. 9. Comparison of the speeds of the nonlinear and FNN network models with wind at 12m/s airspeed.

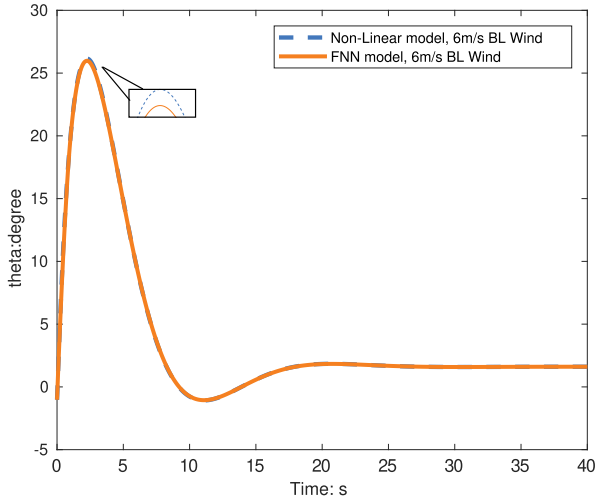

Fig. 10. Comparison of the pitch angles of the nonlinear and FNN Network models with wind at 12m/s airspeed.

especially for reinforcement learning control process, which makes the training process of the DDPG controller difficult.

In the equation above, the parameters a_1 and k_i are the converge speed parameters, where a_1 is $(180/\pi)^3$ to ensure that the attitude errors reach within 1 degree, while k_i set to 16 to satisfy both settling time and overshoot.Because the parameter a_1 is set to ensure that the attitude angle error no longer exceed 1°, while the parameters k_i ensure a suitable converge speed.} The training parameters α_μ and β_Q are set to 10^{-3}. Also, the max episode of the DDPG reinforcement learning is set to 10000. Each episode has a maximum of 4000 step with a 0.01s stepsize. The final reward is a sum of $R(x)$ in all stepsize, and a large reward penalty is applied for unstable episode. The suitable episode reward is set to 10^6. The state s_t is set to the angular error of attitude and the angular rates, with dimension 6, while the action a_t is set to the diagonal parameters of the matrix P_1 in the sliding surface in (9) with a dimension 4.

Wind disturbance is introduced to investigate the disturbance rejection performance. The wind disturbance is based on the boundary layer model. According to [71], the wind speed can be described as

$$V_w = V_{w9.15}\frac{H^{0.2545} - 0.4097}{1.3470} \tag{34}$$

In the equation, $V_{w9.15}$ is the wind speed at a height of 9.15 m. A comparison of the disturbance rejection performances of the proposed method and the linearized model are shown in 7 and 8. The comparison shows that the setting time of the FNN network model system is 10.4 seconds, and the settling time of the linearized model is 11.4 seconds. This is because the proposed method models the FNN network, which has more information than the traditional linearized model. Additionally, the aircraft has a working point different from the linearized point to the linearized model, and the proposed method can model it properly.

The comparison between the nonlinear model and the FNN based model is also shown in Fig.9 and Fig. 10. The previous comparison is based at 16m/s airspeed, and this comparison

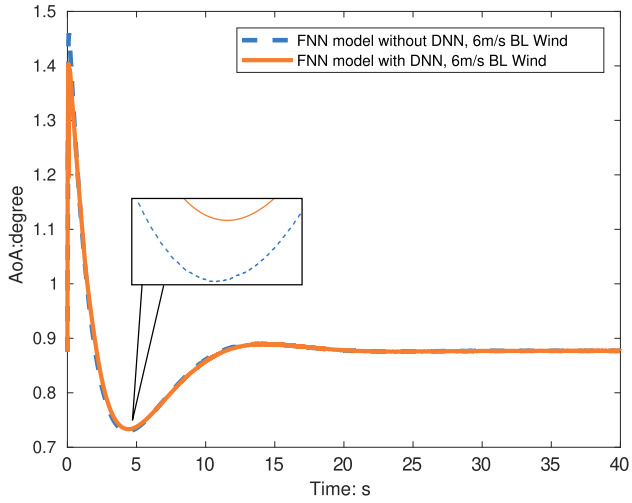

Fig. 11. Angle of Attack comparisons between FNN network modeling with and without a DNN.

is based at 12m/s airspeed. From both comparison, it can be found that in these cases, the overshoot and settling time of the FNN based model is smaller than the nonlinear model, the settling time decrease is 2 seconds, and the overshoot decrease is 1%. The reason is that the FNN network has the ability for unknown parameters at any conditions. It can also replicate the response that the nonlinear model not able to formulate the model at dynamics, which decreases the settling time and overshoot of the aircraft itself.

A DNN is applied to prove the effectiveness of the proposed network. The network training data are the wind effect responses generated from the linearized model. In the second layer, the fully connected layer changes the 8 inputs to the 10 nodes in the deep network. The fully connected layer in layer 5 changes the 10 nodes of the deep network into one layer into only one output. The training is based on the Adam training method. The maximum training epoch is set to 181,440. After the training is complete, the DNN is applied to measure the

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

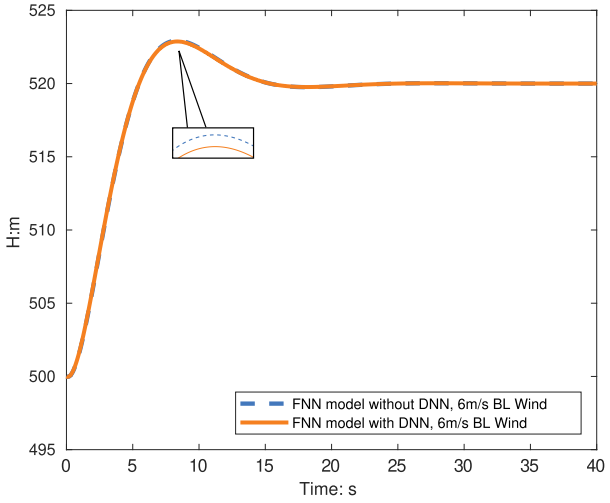

Fig. 12. Height comparison between FNN network modeling with and without a DNN.

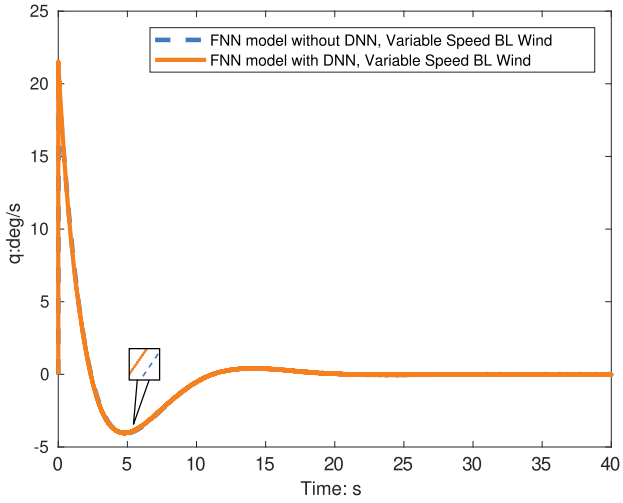

Fig. 14. Comparison of the pitch rate in variable winds without and with a DNN.

TABLE IV
THE TISE COMPARISON BETWEEN THE MODEL WITH AND WITHOUT A DNN

TISE	V	α	q	θ	H
noDNN	0.0011	0.1853	0.1285	5.6286	2.006×10^8
withDNN	0.0017	0.1853	0.1223	5.6094	2.005×10^8

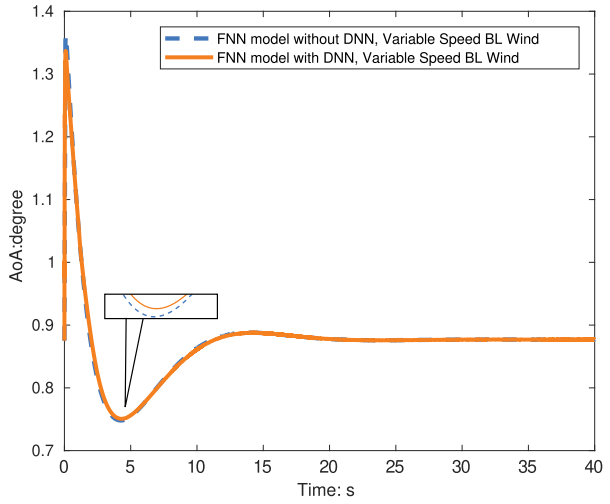

Fig. 13. Comparison of angle of attack in variable winds without and with a DNN.

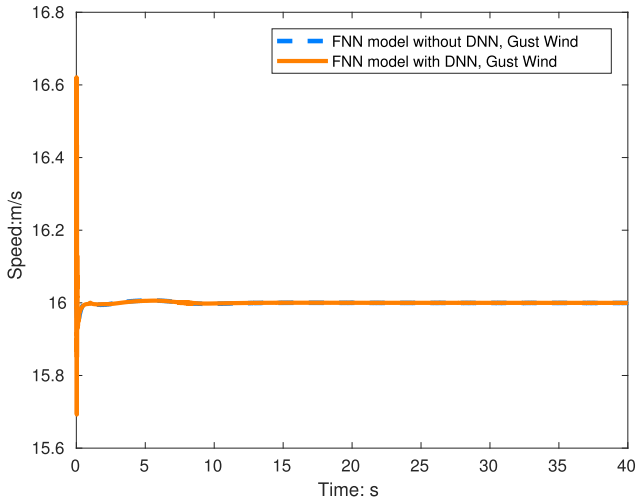

Fig. 15. Comparison of the wind gust speed in FNN network modeling without and with a DNN.

wind disturbance. The control results are shown in Fig. 11 and Fig. 12.

Fig. 11 and 12 show that for the FNN network-based model, the settling time decreases by 1.5 seconds when the deep network is applied, and overshooting decreases by 2% because the wind disturbance could be modeled into the FNN network instead of the derivative equation illustrated in (22).

The time integral squared error (TISE) was applied to estimate the performance of the proposed method [79]. This error is based on a time-varying error. The faster the convergence speed is, the smaller the TISE. Thus, this error is a good identifier for the state response. The TISE with the and without the DNN are illustrated in Table IV.

The TISE comparison shows that the TISE of the model without the DNN is much larger than the model with the deep network. This means that the DNN can decrease wind disturbances. However, it is not significant. The reason is that the FNN network previously modeled the wind disturbance.

More types of wind disturbances were applied to prove the effectiveness of the proposed method. First, the wind speed increases from 4 m/s to 6 m/s at 9.5 m in 20 s, and the results are shown in Fig. 13 and Fig. 14.

According to Fig. 13 and Fig. 14, the settling times of the model without and with the DNN are 11.5 seconds, and 11.2 seconds, respectively. This proves that the wind disturbance was already mitigated by the offline FNN network.

In addition, wind gusts were also considered. The aircraft is affected by the wind gust at 5 seconds; the gust lasted for 10 seconds. The comparison results are shown in Fig. 15 and 16.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

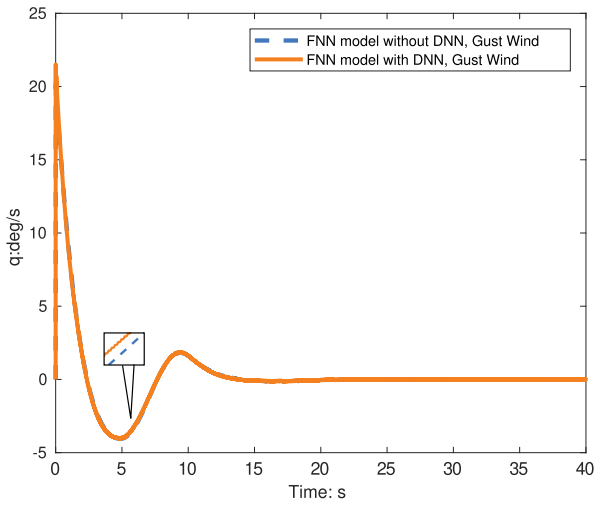

Fig. 16. Comparison of the pitch rate in wind gusts on FNN network modeling without and with a DNN.

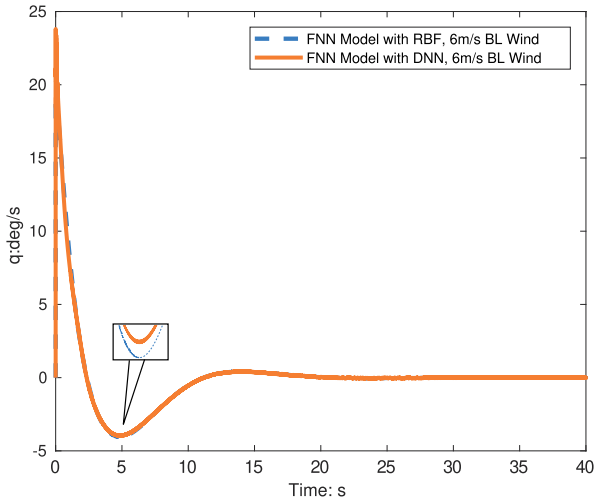

Fig. 17. Comparison of pitch rate in FNN network modeling with RBF and DNN.

According to Fig. 15 and Fig. 16, the settling time of the model without and with the DNN is 11.5 seconds, and 10.7 seconds for gust wind, respectively. Additionally, for overshooting, the model without and with a deep network is 15% and 10%, respectively. This means that the DNN can extract features from wind disturbances even though the model is constructed from FNN networks.

The DNN disturbance observer has been applied to compare with the RBF network observer [79] as well. The compare results are given in Fig. 17 and Fig. 18. From the comparison, it can be found clearly that, in the constant wind environment, the settling time of the model with DNN is 9.9 seconds, while the model with RBF network is 10.2 seconds. The reason is that the DNN has 6 layers, while the RBF has only 3 layers. When extracting features, the DNN network can extract more features than the RBF network, which increses the anti-wind disturbance performance of the aircraft.

Also, the simulation results with DDPG agent in the variable wind speed condition is given in Fig.19 and Fig.20. From

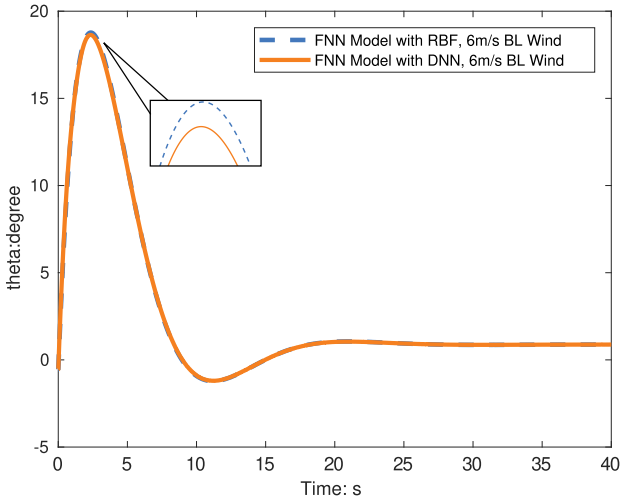

Fig. 18. Comparison of pitch angle in FNN network modeling with RBF and DNN.

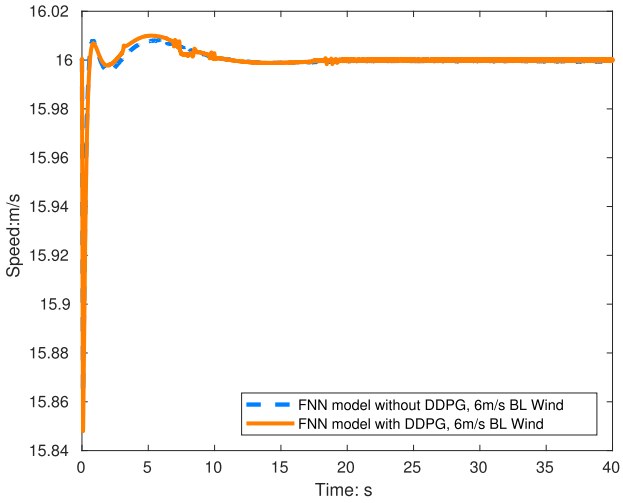

Fig. 19. Airspeed comparison between the model with or without DDPG agent in windy environments.

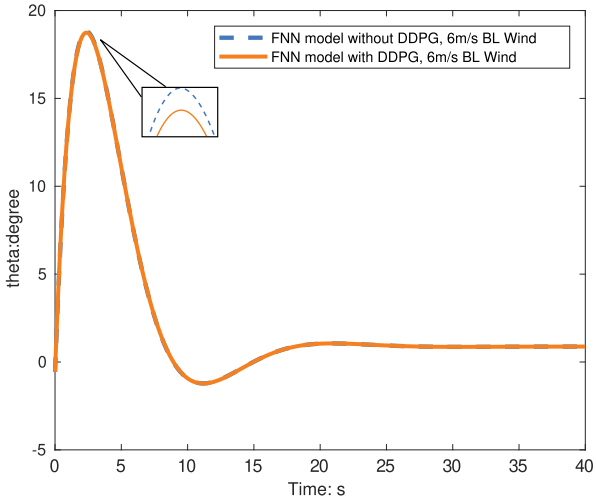

Fig. 20. Pitch angle comparison between the model with or without DDPG agent in windy environments.

the comparison, it can be seen clearly that the response with DDPG agent has a 2 seconds settling time decreasement and the overshooting decrease a 2% as well. The reason is that

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

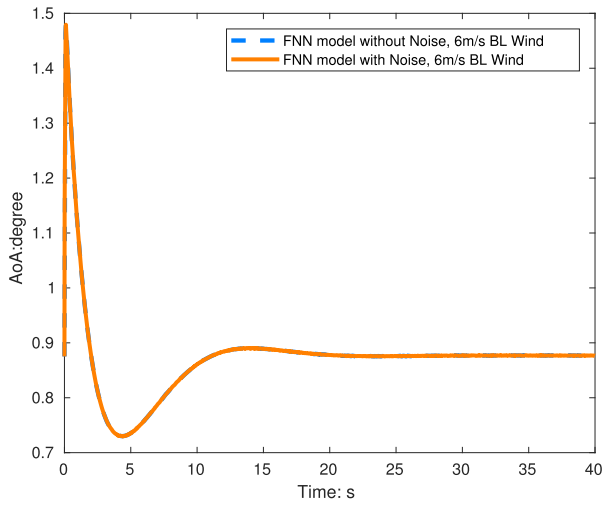

Fig. 21. Comparison of the angle of attack with noise on the FNN network model.

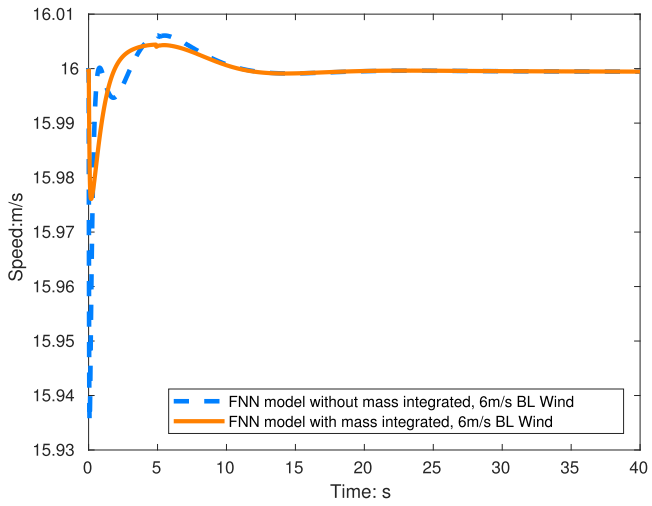

Fig. 23. The results comparison of the airspeed of the mass variation without and with mass integration to the FNN network model.

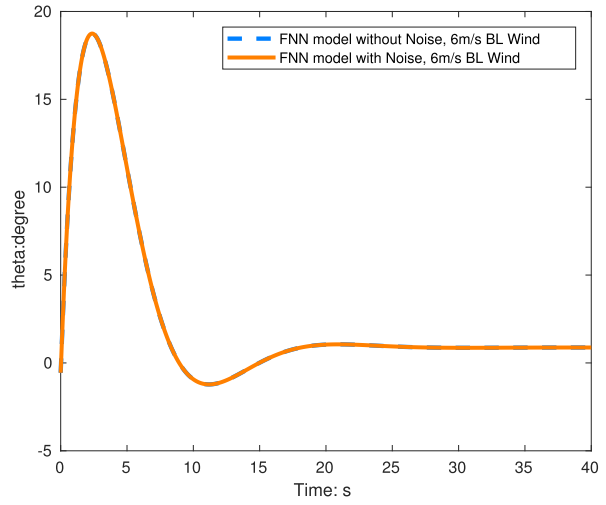

Fig. 22. The results comparison of the pitch angle with noise on the FNN network model.

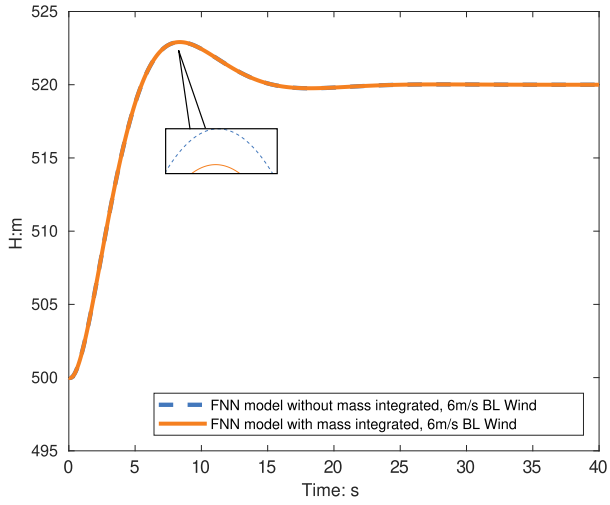

Fig. 24. The results comparison of the height of the mass variation without and with mass integration to the FNN network model.

the DDPG agent has the ability for calculating the optimized result. Besides, the DDPG agent can find the optimized control parameters, which means that with the wind disturbance, the optimized controller can be found by the DDPG agent automatically. This means that in the windy environments, the DDPG agent can find the optimized controller parameters, and the overshoot and settling time decreases significantly.

Additionally, a model with feedback errors is considered. The output of the model is supplemented with 30% noise. These results are compared to those of the model without noise addition. As illustared as follows:

The compared results are illustrated in Fig. 21 and Fig. 22. The comparison shows that with noise integrated into the aircraft, a small shift in the response occurs. Since the aircraft is modeled from the FNN network, the error is not significant. The reason is that FNN networks can measure any function with high robustness. Since the noise is sufficiently small, the shift is not significant. This proves the robustness of the proposed method.

The mass variation application is also considered in the work. The mass of the aircraft decrease from 0.5kg to 0.4kg. These results are compared to the model with mass varying integrated to the FNN network. As illustrated in Fig.23 and Fig.24.

In the proposed work, the DDPG agent is compared to the model without DDPG agent. The results are given in Fig. 25, and Fig. 25. From the simulation, it can be found clearly that by applying the DDPG agent, even in the environment without wind, the response with DDPG agent is better than the model without DDPG agent, with a 1% overshooting improvement and a 0.5 seconds settling time advance. The reason is that the DDPG agent has the ability for automatically calculating the optimized control parameters, which improves the response of the aircraft.

From the comparison, it can be found clearly that in the mass varying applications, with the mass integrated to the FNNs, there is a significant response overshoot and settling time boost. In this case, the mass variation is only 0.1kg, but,

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

TABLE V
COMPARISON OF THE METHOD IN [60], [61], [80] AND [81] AND OUR METHOD IN TISE

TISE	V	α	q	θ	H
Method in [80]	0.0482	0.1861	0.3274	1.0014	1.9993×10^8
Method in [81]	0.0316	0.1862	0.2734	0.7975	1.9993×10^8
Method in [60]	0.0266	0.1863	0.3572	1.0300	2.0019×10^8
Method in [61]	0.2041	0.1862	0.4897	4.3991	1.9992×10^8
Proposed Method	0.0180	0.8929	0.2884	3.2277	2.0009×10^8

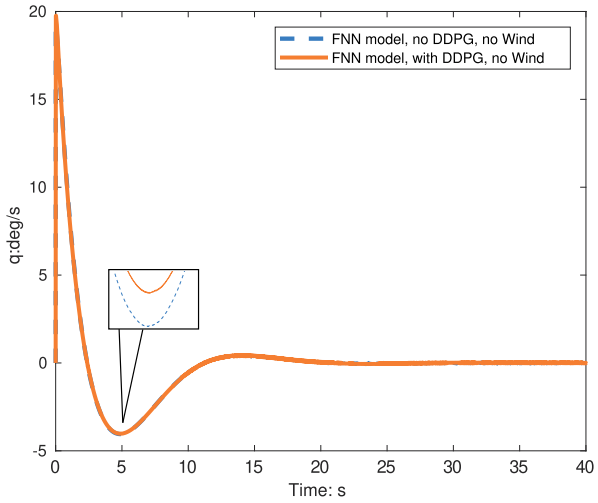

Fig. 25. Pitch rate comparison between the model with or without DDPG agent in normal environments.

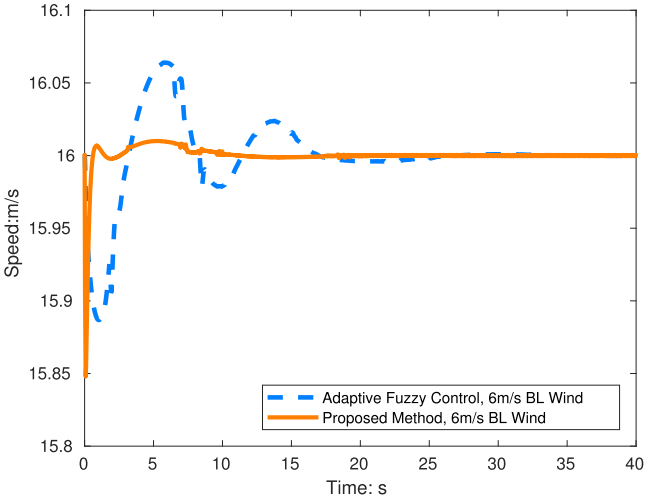

Fig. 27. The comparison of the speed of the proposed method and method in [80].

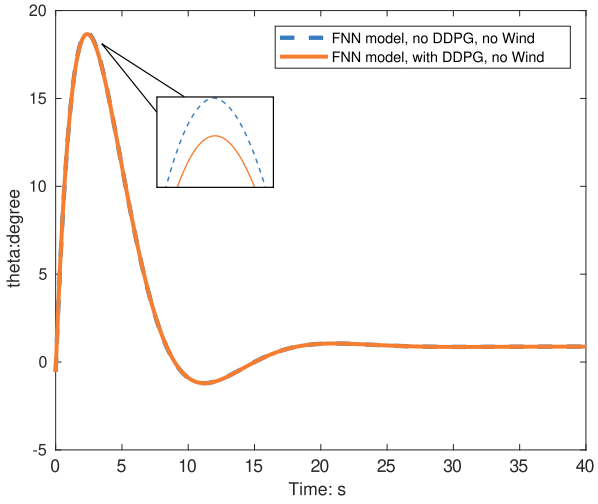

Fig. 26. Comparison between the model with or without DDPG agent in normal environments.

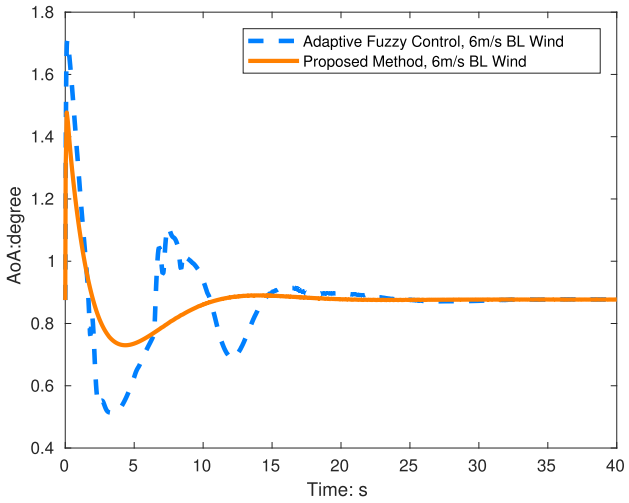

Fig. 28. Comparison of the AoAs of the proposed method and the methods in [80].

with mass integrated, the settling time decreases 0.2 seconds, and the overshoot decrease 0.5%. The reason is that the FNN network can measuring any functions within a desired error rate, precisely modeling the weight changing of the flying wing aircraft.

The proposed method has been compared to the control strategies introduced in [60], [61], [80], and [81]. The method introduced in [80] has applied an adaptive fuzzy sliding mode control strategy and the authors in [81] have introduced a method based on adaptive fuzzy sliding mode based on the Nussbaum function. Besides, the method introduced in [60] has applied an PID based SAC deep reinforcement learning strategy, and the authors in [61] have introduced a method based on TD3 reinforcement learning strategy with nonlinear dynamic. The control results are shown in Fig.27, Fig.28,

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

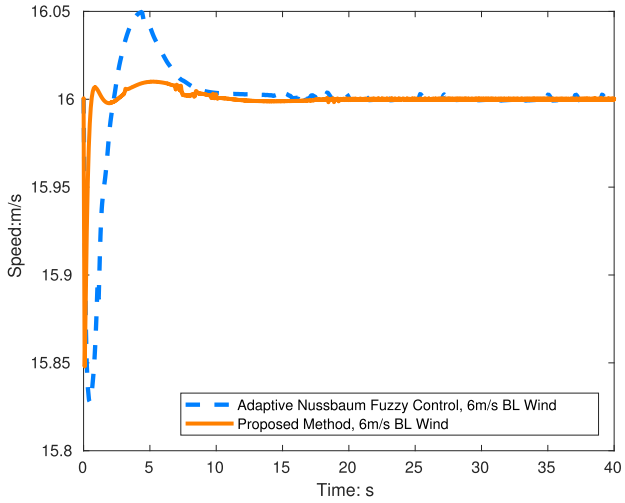

Fig. 29. The comparison of the speed of the proposed method and the method in [81].

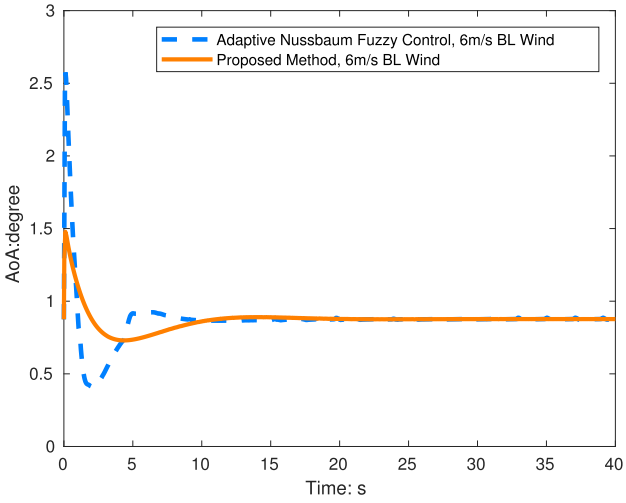

Fig. 30. Comparison of the AoAs of the proposed method and the methods in [81].

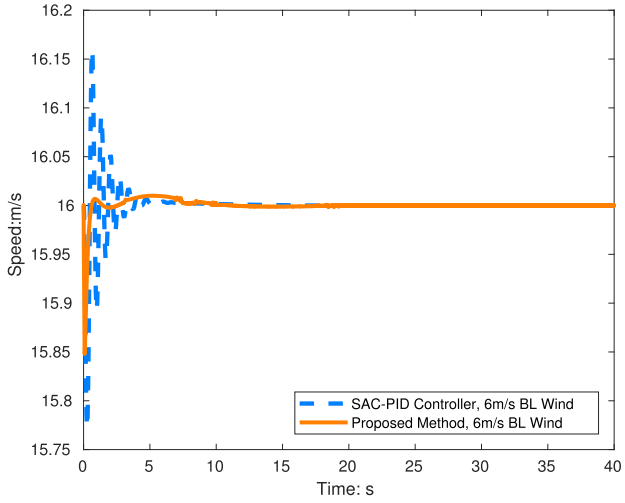

Fig. 31. The comparison of the speed of the proposed method and method in [60].

Fig.29, Fig.30, Fig.31, Fig.32, Fig.33, and Fig.34. The TISE comparison is illustrated in Table V.

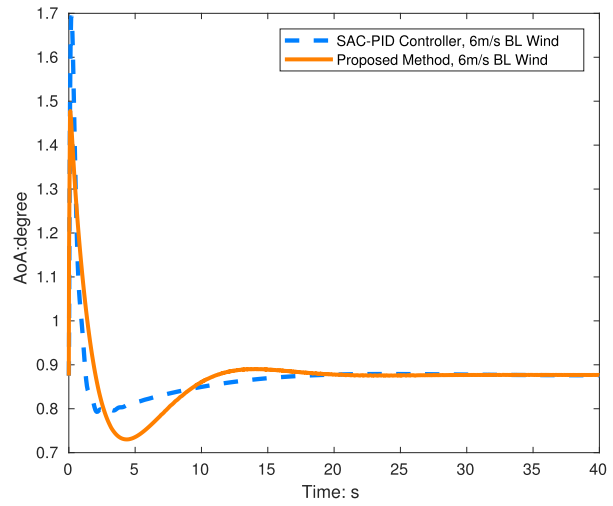

Fig. 32. Comparison of the AoAs of the proposed method and the methods in [60].

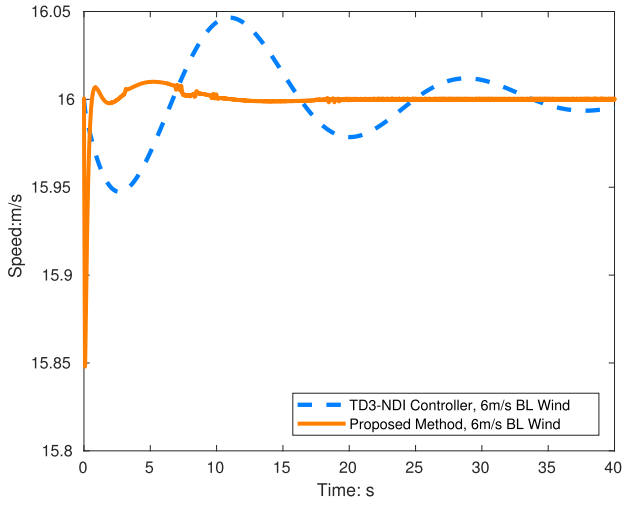

Fig. 33. The comparison of the speed of the proposed method and the method in [61].

The comparison clearly shows that the proposed method performs better than the method applied in [80]. This is because the proposed method introduces a FNN network to measure the aerodynamic parameters, the attitude response is faster. However, the overshoot and settling time is similar to proposed method. Since the method in [80] uses a fuzzy controller, it has a better performance. Compared to the method applied in [81], the settling time and overshoot of the proposed method are both greater than the method in [81], the reason is that in [81], the nussbaum function has a significant in dynamic performance improvement, which decreases both settling time and overshoot at the same time. Also, the proposed method has a better performance than [61], including that the attitude response is faster and the overshoot and settling time is better than the proposed method. Since the method in [60] uses the reinforcement learning strategy to adjust the PID parameters, which is a bit harder to be adjusted than sliding mode. Compared to the method applied in [61], the settling time and overshoot of the proposed method are

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

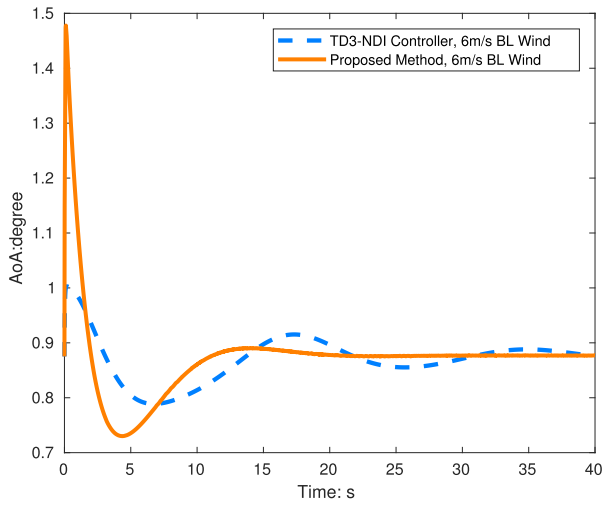

Fig. 34. Comparison of the AoAs of the proposed method and the methods in [61].

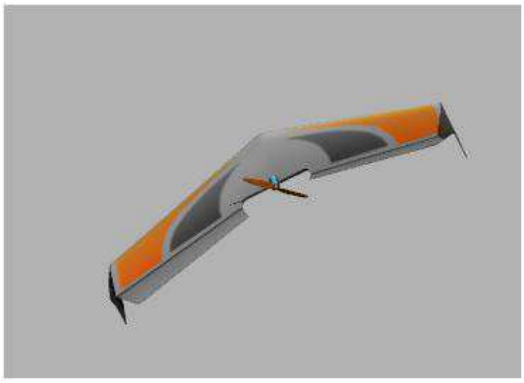

Fig. 35. The Zephyr delta Wing model used for SITL.

both greater than the method in [61], the reason is that in [61], the method uses the nonlinear inverse model based function, and the reinforcement learning is based on the attitude angle loop, while the proposed method is based on the angular rate loop with a sliding surface, which is much more robust than the method in [61].

However, the FNN network modeling has limitations. First of all, it is an offlined modeling strategy, which means that it can only modeling the aircraft data in the range of training data. For the outside of training data, or the aircraft itself is changed, there is error in the modeling process. This is the disadvantage of the proposed method, like other offlined modeling. However, for the points within the modeling frame, the proposed method shows a very good performance. Besides, for a different aircraft, the FNN network should be modeled separately.

B. Software in the Loop

The software in the loop (SITL) was also applied for the proposed control strategies. The SITL is based on Gazebo version 11.12 with the PX4 framework. The aircraft is the Zephyr delta wing model shown in Fig.35. the aircraft parameters is illustrated in Table. VI, and VII. the aerodynamic parameters are given in Table.VIII.

TABLE VI
PARAMETER OF THE ZEPHYR DELTA WING

Wingspan	Chores	MAC	Aspect Ratio	Surface
$2b$	c_0	c_A	λ	S
1521mm	447.54mm	311.9mm	3.4591	$0.6688m^2$

TABLE VII
MASS AND MOMENTS INERTIAS OF THE ZEPHYR DELTA WING

Parameters	Values
I_{xx}	0.387382402 kgm^2
I_{yy}	0.083137104 kgm^2
I_{zz}	0.469845106 kgm^2
I_{xz}	0.000641038 kgm^2
m	0.50kg

TABLE VIII
AERODYNAMIC PARAMETERS OF ZEPHYR DELTA WING

Parameters	Values	Parameters	Values
C_{D0}	0.019732	C_{L0}	0.01417462
C_{m0}	0.000407	$C_{D\alpha}$	0.0540
$C_{L\alpha}$	3.3302	C_{ma}	0.04085
$C_{Y\beta}$	-0.1075	$C_{l\beta}$	-0.0231
$C_{n\beta}$	0.0172	C_{Yp}	-1.13
C_{lp}	-2.3	C_{np}	-0.11
C_{Dq}	0.659	C_{Lq}	1.02
C_{mq}	-0.724	C_{Yr}	-1.25
C_{lr}	-0.289	C_{nr}	-0.8
$C_{L\alpha_{stall}}$	-0.3326	$C_{D\alpha_{stall}}$	0.9204
$C_{m\alpha_{stall}}$	0.1223	$C_{Y\delta a}$	-0.00171
$C_{l\delta a}$	-0.3331	$C_{n\delta a}$	-0.00057
$C_{L\delta e}$	0.10696	$C_{D\delta e}$	-0.000274
$C_{m\delta e}$	-0.025798	$C_{Y\delta r}$	-0.003913
$C_{l\delta r}$	-0.000257	$C_{n\delta r}$	0.001613

The simulation is combined with MATLAB in Linux. The aircraft takes off with the PX4 controller. After it reaches a certain value, the controller switches to the proposed method mentioned above. The method switch condition is the current airspeed and the aircraft is actually taking off, as shown in (35)

$$ext_{controller} = (V \geq V_l \quad and \quad takeoff = true) \tag{35}$$

In the equation above, the condition makes sure that the aircraft is flying properly in the air. This is because that the proposed method is designed on level flight.

The SITL simulation of non-linearized model is illustrated in Fig.36. The FNN network based model without DDPG is illustrated in Fig.37, and the FNN network based model with DDPG is illustrated in Fig.38.

Simulations in windy environments were also applied. In windy environments, a deep network is applied through a selectable key. If the aircraft speed is faster than V_L, the

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

Fig. 36. SITL Simulation based on non-linearized model.

Fig. 37. SITL Simulation based based on FNN Network model.

Fig. 38. SITL Simulation based on FNN Network model with DDPG agent.

control strategy is swapped into the proposed control strategy. Applying the simulation shows that when the control strategy is swapped to the proposed method, the tracking errors heavily decreases. The tracking error based on the FNN network is 1%, while the tracking error based on DDPG controller is 0.5%. While the tracking error based on the PX4 controller is 5%. Also, the settling time and overshoots also decreases. For the non-linearized model, the settling time is 10 seconds, the overshooting is 6%. For the model with FNN network,

TABLE IX
MAE ERROR IN SITL

Channel	PX4	FNN	FNN+RL
x	40.35	35.65	34.55
y	15.35	12.48	11.52
h	3.15	3.75	3.61

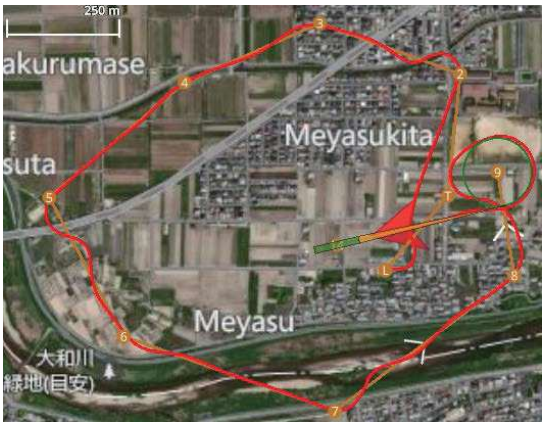

Fig. 39. HITL Simulation based on non-linearized model.

TABLE X
MEMORY USED IN DIFFERENT CONTROL STRATEGIES

Method	Memory Usage
PX4 Base	1.50MB
FNN	1.77MB
FNN+DNN	1.98MB
FNN+RL	1.86MB

the settling time is 9.5 seconds, and the overshooting is 5.5%. For the model based on FNN network with DDPG agent, the settling time is 5, and the overshooting is 5.2%. Since the PX4 controller is nonlinear itself, it can be seen clearly that the FNN network has better performance for modeling a function, and the DDPG agent has the ability to choose the optimized control parameters, at different situations, the result is always chosen from the best response situation from the actor of the DDPG agent. Besides, the MAE(Mean Absolute Error) error are shown in Table. IX.

From the error, it can be found clearly that the MAE of the FNN modeling is smaller than the MAE of PX4 original method. The reason is that the FNN networks have a better modeling performance than the traditional nonlinear modeling control strategies. Besides, the error decreases further when the DDPG Reinforcement Learning is added, the error decrease further. This means that the DDPG agent has the ability to choose the optimized control process that decreases the error and increases the dynamic performance.

C. Hardware In the Loop

To prove the effectiveness of the proposed method in the hardware applications, the hardware in the loop(HITL) is also applied to the proposed method. The hardware itself is the

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

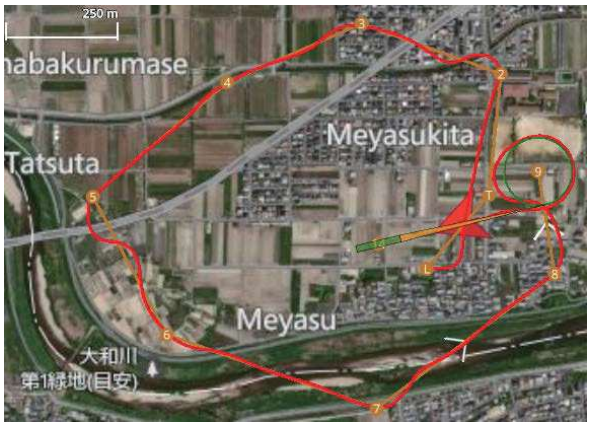

Fig. 40. HITL Simulation based on FNN Network model.

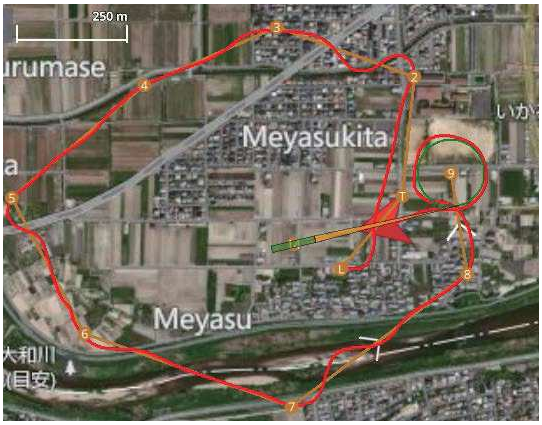

Fig. 41. HITL Simulation based on FNN Network model with DNN state observer.

Fig. 42. HITL Simulation based on FNN Network model with DDPG agent.

CUAV V5 Nano, and the aircraft frame is the Zephyr Delta Wing model mentioned in the SITL simulation, with the baud rate 921600. The simulation parameters are exactly same to SITL. Since the size of ROM in CUAV V5 Nano is 2MB only, and the code with FNN is 2.1MB, some modules of PX4 that does not affect the HITL simulation have been disabled. The simulation results are given as follows:The model without FNN network is illustrated in Fig. 39, the model with FNN network is illustrated in Fig. 40, the model with FNN network and DNN observer is illustrated in Fig. 41, and the model based on the FNN network with DDPG is illustrated in Fig. 42. Note that the size of DNN is compressed to 64 nodes instead of 128 of original simulation due to the limitation of the firmware size of the controller. The memory usage of different methods is shown in Table.X.

TABLE XI
MAE ERROR IN HITL

Channel	PX4	FNN	FNN+DNN	FNN+RL
x	41.77	34.41	36.10	33.96
y	14.65	11.98	12.38	10.73
h	2.25	2.65	3.34	2.58

The HITL simulation is similar to the SITL simulation, and the switch method is based on the current flying status. It can be seen clearly that the tracking error based on PX4 controller is 4.5%, the settling time is 5 seconds, and the overshoot is 6.3%. For the control method based on the FNN network, the tracking error is 0.9%, the settling time is 4.5 seconds, and the overshoot is 5.7%. For the model based on the FNN network with DDPG integration, the tracking error is 0.46%, the settling time is 4.2 seconds, and the overshoot is 4.5%. The MAE error of each method are given in Table. XI.

Similar to the SITL, it the HITL simulation, the same pattern occurs. However, there is a bad value that when adding the DNN, the error increases. This is because that since the DNN always adds the wind disturbance measurement in the flying process, even if there is no wind. But the DNN won't stop estimating, this is the reason that DNN error is higher. To overcome this, the no wind condition should be considered in the future works.

VI. CONCLUSION

In this paper, a novel intelligent modeling and control framework for flying wing aircraft is introduced, designed to achieve higher precision aircraft modeling across diverse and uncertain flight conditions. The core theoretical contribution lies in the integration of FNN networks for global nonlinear system modeling, a DDPG agent for adaptive control optimizing, and a DNN based observer for wind disturbance estimation. This unified framework explicitly addresses critical challenges often overlooked in isolation, including mass variation, parametric uncertainties, and turbulent wind conditions.

Comprehensive simulations and tests demonstrate the efficacy of this integrated approach. The FNN networks prove capable of accurately modeling the aircraft's dynamics across a wide envelope of flight conditions, providing a foundational model for the downstream controller and observer. The DDPG agent successfully learns to optimize control policies in interaction with this model, while the DNN observer effectively estimates and compensates for wind disturbances. The framework's practical viability and robustness are conclusively validated through both SITL and HITL simulations, confirming its performance in realistic scenarios and its successful deployment on operational hardware.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

REFERENCES

[1] D. K. Tiep and Y.-J. Ryoo, "An autonomous control of fuzzy-PD controller for quadcopter," *Int. J. FUZZY Log. Intell. Syst.*, vol. 17, no. 2, pp. 107–113, Jun. 2017.

[2] M. Komnatska, "Optimal flight control system design via LMIs," in *Proc. IEEE 3rd Int. Conf. Methods Syst. Navigat. Motion Control (MSNMC)*, Oct. 2014, pp. 158–160.

[3] Y. Jing, J. Xu, G. M. Dimirovski, and Y. Zhou, "Optimal nonlinear estimation for aircraft flight control in wind shear," in *Proc. Amer. Control Conf.*, Jun. 2009, pp. 3813–3818.

[4] J. Smith, J. Su, C. Liu, and W.-H. Chen, "Disturbance observer based control with anti-windup applied to a small fixed wing UAV for disturbance rejection," *J. Intell. Robotic Syst.*, vol. 88, nos. 2–4, pp. 329–346, Dec. 2017.

[5] V. Martinez, O. Garcia, A. Sanchez, V. Parra, and A. Escobar, "Adaptive backstepping control for a convertible UAV," in *Proc. Workshop Res., Educ. Develop. Unmanned Aerial Syst. (RED-UAS)*, Nov. 2015, pp. 298–307.

[6] J. R. Hervas, E. Kayacan, M. Reyhanoglu, and H. Tang, "Sliding mode control of fixed-wing UAVs in windy environments," in *Proc. 13th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 986–991.

[7] Q. Yang, G. Cai, D. Mao, and X. Lv, "Design of LPV controller for morphing aircraft based on finite-time time-varying sliding mode," in *Proc. 41st Chin. Control Conf. (CCC)*, Jul. 2022, pp. 3579–3584.

[8] C. Du, F. Li, C. Yang, Y. Shi, L. Liao, and W. Gui, "Multiphase-based optimal slip ratio tracking control of aircraft antiskid braking system via second-order sliding-mode approach," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 2, pp. 823–833, Apr. 2022.

[9] Y. Hou, D. Chen, and S. Yang, "Adaptive robust trajectory tracking controller for a quadrotor UAV with uncertain environment parameters based on backstepping sliding mode method," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 4446–4456, 2025.

[10] B. Wang, B. Brogliato, V. Acary, A. Boubakir, and F. Plestan, "Experimental comparisons between implicit and explicit implementations of discrete-time sliding mode controllers: Toward input and output chattering suppression," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 2071–2075, Sep. 2015.

[11] E. W. Metekia, W. A. Asfaw, C. M. Abdissa, and L. N. Lemma, "Control of a fixed wing unmanned aerial vehicle using a robust fractional order controller," *Sci. Rep.*, vol. 15, no. 1, p. 19954, Jun. 2025.

[12] S. Liu, B. Jiang, Z. Mao, Y. Zhang, and J. Huang, "Adaptive fractional-order fault-tolerant coordinated tracking control of heterogeneous multiagent systems against multiple faults under deception attacks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 61, no. 2, pp. 1860–1870, Apr. 2025.

[13] Y. Deia, M. Kidouche, and A. Ahriche, "Fully decentralized fuzzy sliding mode control with chattering elimination for a quadrotor attitude," in *Proc. 4th Int. Conf. Electr. Eng. (ICEE)*, Dec. 2015, pp. 1–6.

[14] K. V. Kumar, T. A. Kumar, and V. Ganesh, "Chattering free sliding mode controller for load frequency control of multi area power system in deregulated environment," in *Proc. IEEE 7th Power India Int. Conf. (PIICON)*, Nov. 2016, pp. 1–6.

[15] Z.-R. Feng, R.-Z. Sha, and Z.-G. Ren, "A chattering-reduction sliding mode control algorithm for affine systems with input matrix uncertainty," *IEEE Access*, vol. 10, pp. 58982–58996, 2022.

[16] L. Zhou, Z. Li, H. Yang, C. Tan, and Y. Fu, "Adaptive terminal sliding mode control for high-speed EMU: A MIMO data-driven approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 1970–1983, 2025.

[17] V. Utkin, "Discussion aspects of high-order sliding mode control," *IEEE Trans. Autom. Control*, vol. 61, no. 3, pp. 829–833, Mar. 2016.

[18] G. P. Incremona, M. Rubagotti, M. Tanelli, and A. Ferrara, "A general framework for switched and variable gain higher order sliding mode control," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1718–1724, Apr. 2021.

[19] Y. Long, C. Yao, and E.-Z. Song, "Design and experimental analysis of an adaptive second-order fast non-singular terminal sliding mode controller for electronic throttle with disturbance," *IEEE Access*, vol. 11, pp. 57854–57866, 2023.

[20] E. A. Teklu and C. M. Abdissa, "Genetic algorithm tuned super twisting sliding mode controller for suspension of Maglev train with flexible track," *IEEE Access*, vol. 11, pp. 30955–30969, 2023.

[21] H. Saied, A. Chemori, M. Bouri, M. E. Rafei, and C. Francis, "FeedForward super-twisting sliding mode control for robotic manipulators: Application to PKMs," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 3167–3184, Aug. 2023.

[22] Y. Bi, F. Wang, P. Ding, T. Wang, and J. Qiu, "Multivariable adaptive super-twisting sliding mode resilient control for uncertain nonlinear CPSs against actuator and sensor attacks," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 9039–9048, 2025.

[23] B. Wang, T. Wang, Y. Yu, and D. Xu, "Second-order terminal sliding-mode speed controller for induction motor drives with nonlinear control gain," *IEEE Trans. Ind. Electron.*, vol. 70, no. 11, pp. 10923–10934, Nov. 2023.

[24] S. Lian, Y. Zhu, W. Meng, K. Shao, and H. Li, "Adaptive fault-tolerant control for quadrotor based on the second-order fast nonsingular terminal sliding mode control," *IEEE Trans. Ind. Electron.*, vol. 72, no. 8, pp. 8322–8332, Aug. 2025.

[25] G. A. Garcia, S. S. Keshmiri, and T. Stastny, "Robust and adaptive nonlinear model predictive controller for unsteady and highly nonlinear unmanned aircraft," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1620–1627, Jul. 2015.

[26] T. Yue, X. Zuo, L. Wang, J. Geng, and H. Zhang, "Similarity relations of PID flight control parameters of scaled-model and full-size aircraft," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 4, pp. 2950–2960, Aug. 2022.

[27] J. Zhang, I. Roumeliotis, and A. Zolotas, "Nonlinear model predictive control-based optimal energy management for hybrid electric aircraft considering aerodynamics-propulsion coupling effects," *IEEE Trans. Transport. Electrific.*, vol. 8, no. 2, pp. 2640–2653, Jun. 2022.

[28] H. Ren, B. Hou, G. Zhou, L. Shen, C. Wei, and Q. Li, "Variable pitch active disturbance rejection control of wind turbines based on BP neural network PID," *IEEE Access*, vol. 8, pp. 71782–71797, 2020.

[29] J. Zheng, Y. Wang, C. Lu, and D. Tian, "Joint space based force sensorless bilateral control with BP neural network gravity compensation for 6-PSS parallel actuator," in *Proc. 15th Int. Conf. Human Syst. Interact. (HSI)*, Jul. 2022, pp. 1–6.

[30] M. Wang, X. Dong, and X. Ren, "Improved BP neural network based active disturbance rejection control for magnetic sensitivity calibration system," in *Proc. IEEE 12th Data Driven Control Learn. Syst. Conf. (DDCLS)*, May 2023, pp. 1002–1007.

[31] H. Ma, J. Zhou, J. Zhang, and L. Zhang, "Research on the inverse kinematics prediction of a soft biomimetic actuator via BP neural network," *IEEE Access*, vol. 10, pp. 78691–78701, 2022.

[32] J. Zhang et al., "Lifetime extension approach based on the Levenberg–Marquardt neural network and power routing of DC–DC converters," *IEEE Trans. Power Electron.*, vol. 38, no. 8, pp. 10280–10291, Aug. 2023.

[33] U. C. Hasar, H. Ozturk, M. Ertugrul, J. J. Barroso, and O. M. Ramahi, "Artificial neural network model for evaluating parameters of reflection-asymmetric samples from reference-plane-invariant measurements," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–8, 2023.

[34] K. Yong, Q. Wu, and M. Chen, "Wind estimation-based robust flight control for UAV with active maneuverability limit," in *Proc. IEEE 28th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2019, pp. 682–687.

[35] J. S. Lorenz, "Evaluation of different wind estimation methods in flight tests with a fixed-wing UAV," Robotics Institude, Pittsbourg, PA, USA, Tech. Rep. CMU-RI-TR-18-03, 2018.

[36] X. Liang, Y. Wang, H. Yu, Z. Zhang, J. Han, and Y. Fang, "Observer-based nonlinear control for dual-arm aerial manipulator systems suffering from uncertain center of mass," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 1984–1995, 2025.

[37] M. K. Al-Sharman, Y. Zweiri, M. A. K. Jaradat, R. Al-Husari, D. Gan, and L. D. Seneviratne, "Deep-learning-based neural network training for state estimation enhancement: Application to attitude estimation," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 1, pp. 24–34, Jan. 2020.

[38] T. Matassini, H. Shin, A. Tsourdos, and M. Innocenti, "Adaptive control with neural networks-based disturbance observer for a spherical UAV," in *Proc. IEEE Conf. Int. Fed. Autom. Control(IFAC)*, Jun. 2016, vol. 49, no. 17, pp. 308–313.

[39] T. Lan et al., "Event-triggered fixed-time sliding mode control for lip-reading-driven UAV: Disturbance rejection using wind field optimization," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 9090–9103, 2025.

[40] J. K. Lee, "A parallel attitude-heading Kalman filter without state-augmentation of model-based disturbance components," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 7, pp. 2668–2670, Jul. 2019.

[41] G. Tian, J. Tan, B. Li, and G. Duan, "Optimal fully actuated system approach-based trajectory tracking control for robot manipulators," *IEEE Trans. Cybern.*, vol. 54, no. 12, pp. 7469–7478, Dec. 2024.

[42] X. Xu and B. Li, "PDE-based observation and predictor-based control for linear systems with distributed infinite input and output delays," *Automatica*, vol. 170, Dec. 2024, Art. no. 111845.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

[43] Q. Meng, Q. Ma, and Y. Shi, "Adaptive fixed-time stabilization for a class of uncertain nonlinear systems," *IEEE Trans. Autom. Control*, vol. 68, no. 11, pp. 6929–6936, Nov. 2023.

[44] D. Yu, S. Ma, Y.-J. Liu, Z. Wang, and C. L. P. Chen, "Finite-time adaptive fuzzy backstepping control for quadrotor UAV with stochastic disturbance," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 1335–1345, Apr. 2024.

[45] M. Li, Y. Chen, M. Zhang, and H. Song, "Adaptive NN observer-based synthesize strategy for connected nonlinear system," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 14798–14807, 2025.

[46] B. Li, Y. Li, P. Yang, and X. Zhu, "Adaptive neural network-based fault-tolerant control for quadrotor-slung-load system under marine scene," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 681–691, Jan. 2024.

[47] N. Wanore Madebo, C. Merga Abdissa, and L. Negash Lemma, "Enhanced trajectory control of quadrotor UAV using fuzzy PID based recurrent neural network controller," *IEEE Access*, vol. 12, pp. 190454–190469, 2024.

[48] T. Beharu Arega, Y. Mersha Tesfa, and C. M. Abdissa, "Three-wheeled mobile robot trajectory tracking control using nonlinear PID controller based neural network combined with backstepping controller," *IEEE Access*, vol. 13, pp. 100167–100182, 2025.

[49] S. Shen, J. Xu, P. Chen, and Q. Xia, "Adaptive neural network extended state observer-based finite-time convergent sliding mode control for a quad tiltrotor UAV," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 5, pp. 6360–6373, May 2023.

[50] G. Yu, J. Reis, and C. Silvestre, "Quadrotor neural network adaptive control: Design and experimental validation," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2574–2581, May 2023.

[51] N. W. Madebo, "Enhancing intelligent control strategies for UAVs: A comparative analysis of fuzzy logic, fuzzy PID, and GA-optimized fuzzy PID controllers," *IEEE Access*, vol. 13, pp. 16548–16563, 2025.

[52] Z. Yu, Y. Li, M. Lv, B. Pei, and A. Fu, "Event-triggered adaptive fuzzy fault-tolerant attitude control for tailless flying-wing UAV with fixed-time convergence," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 4858–4869, Apr. 2024.

[53] Y. Ren, Y. Sun, Z. Liu, and H.-K. Lam, "Parameter-optimization-based adaptive fault-tolerant control for a quadrotor UAV using fuzzy disturbance observers," *IEEE Trans. Fuzzy Syst.*, vol. 33, no. 2, pp. 593–605, Feb. 2025.

[54] F. T. Yareshe, N. W. Madebo, C. M. Abdissa, and L. N. Lemma, "Trajectory tracking of fixed-wing UAV using ANFIS-based sliding mode controller," *IEEE Access*, vol. 13, pp. 61986–62003, 2025.

[55] H. D. Choi, K. S. Kim, P. Shi, and C. K. Ahn, "Adaptive neuro-fuzzy sliding mode tracking for quadrotor UAVs," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 16322–16333, 2025.

[56] X. Chen, M. Zhang, Z. Wu, L. Wu, and X. Guan, "Model-free load frequency control of nonlinear power systems based on deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 20, no. 4, pp. 6825–6833, Apr. 2024.

[57] G. Hao et al., "A deep deterministic policy gradient approach for vehicle speed tracking control with a robotic driver," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2514–2525, Jul. 2022.

[58] X. Mao, G. Wu, M. Fan, Z. Cao, and W. Pedrycz, "DL-DRL: A double-level deep reinforcement learning approach for large-scale task scheduling of multi-UAV," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 1028–1044, 2025.

[59] B. Ma et al., "Deep reinforcement learning of UAV tracking control under wind disturbances environments," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[60] E. Bøhn, E. M. Coates, D. Reinhardt, and T. A. Johansen, "Data-efficient deep reinforcement learning for attitude control of fixed-wing UAVs: Field experiments," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 3168–3180, Mar. 2024.

[61] W. Hu et al., "TD3 agent-based nonlinear dynamic inverse control for fixed-wing UAV attitudes," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–12, 2025.

[62] J. Tan, S. Xue, H. Li, Z. Guo, H. Cao, and B. Chen, "Hierarchical safe reinforcement learning control for leader-follower systems with prescribed performance," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 19568–19581, 2025.

[63] J. Tan, S. Xue, Q. Guan, K. Qu, and H. Cao, "Finite-time safe reinforcement learning control of multi-player nonzero-sum game for quadcopter systems," *Inf. Sci.*, vol. 712, Sep. 2025, Art. no. 122117.

[64] H. Xu, X. Lv, X. Wang, Z. Ren, N. Bodla, and R. Chellappa, "Deep regionlets: Blended representation and deep learning for generic object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1914–1927, Jun. 2021.

[65] M. R. U. Saputra, C. X. Lu, P. P. B. de Gusmao, B. Wang, A. Markham, and N. Trigoni, "Graph-based thermal–inertial SLAM with probabilistic neural networks," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1875–1893, Jun. 2022.

[66] J. Shi, B. Foggo, and N. Yu, "Power system event identification based on deep neural network with information loading," *IEEE Trans. Power Syst.*, vol. 36, no. 6, pp. 5622–5632, Nov. 2021.

[67] Y. Zhen and M. Hao, "Aircraft control method based on deep reinforcement learning," in *Proc. IEEE 9th Data Driven Control Learn. Syst. Conf. (DDCLS)*, Nov. 2020, pp. 912–917.

[68] Y. Wang, S. Fang, J. Hu, and D. Huang, "A novel active disturbance rejection control of PMSM based on deep reinforcement learning for more electric aircraft," *IEEE Trans. Energy Convers.*, vol. 38, no. 2, pp. 1461–1470, Jun. 2023.

[69] H. Ma, B. Song, Y. Pei, and Z. Chen, "Efficiency change of control surface of a biomimetic wing morphing UAV," *IEEE Access*, vol. 8, pp. 45627–45640, 2020.

[70] V. Prisacariu, I. Circiu, and M. Boscoianu, "*Flying wing aerodynamic analysis*," vol. 2, pp. 31–35, Jan. 2012.

[71] M. Rauw, "FDC 1.2—A simulink toolbox for flight dynamics and control analysis," Inholland Univ., Haarlem, The Netherland, Tech. Rep., 2010.

[72] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *Int. J. Control*, vol. 76, nos. 9–10, pp. 924–941, Jan. 2003.

[73] Y. Song, W. Zhao, Z. Liu, J. Han, and Q. Dang, "Nonlinear sliding mode control of flying wing aircraft under wind disturbance," *Meas. Sci. Technol.*, vol. 36, no. 2, Jan. 2025, Art. no. 026206.

[74] D. Shukla, H. Benyamen, S. Keshmiri, and N. M. Beckage, "Reinforcement learning-based evolving flight controller for fixed-wing uncrewed aircraft," *IEEE Trans. Control Syst. Technol.*, vol. 33, no. 3, pp. 872–886, May 2025.

[75] A. A. Lambregts, "TECS generalized airplane control system design C an update," in *Proc. 2nd CEAS Specialist Conf. Guid., Navigat. Control (EuroGNC)*, 2014, pp. 1284–1303.

[76] F. M. Zegers, R. Sun, G. Chowdhary, and W. E. Dixon, "Distributed state estimation with deep neural networks for uncertain nonlinear systems under event-triggered communication," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 3107–3114, May 2023.

[77] A. Distante and C. Distante, *RBF, SOM, Hopfield, and Deep Neural Networks*. Cham, Switzerland: Springer, 2020, pp. 193–260.

[78] K. Madsen, H. B. Nielsen, and O. Tingleff, *Methods for Non-Linear Least Squares Problems* (Informatics and Mathematical Modelling), 2nd ed., Technical University of Denmark, Apr. 2004.

[79] Y. Song, Z. Liu, J. Han, W. Zhao, and N. Li, "Coordinated turn of fixed-wing aircraft under wind interface on integral backstepping," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–18, 2021.

[80] W. Li, X. Han, Y. Zhi, B. Wang, L. Liu, and H. Fan, "Adaptive finite-time incremental backstepping fault-tolerant control for flying-wing aircraft with state constraints," *Aerosp. Sci. Technol.*, vol. 147, Apr. 2024, Art. no. 108968.

[81] W. Yang, X. Wen, L. Mo, M. Lv, Z. Yu, and M. Wu, "Event-triggered fixed-time fault-tolerant attitude control for the flying-wing UAV using a nussbaum-type function," *Aerosp. Sci. Technol.*, vol. 152, Sep. 2024, Art. no. 109336.

Yuecheng Song is currently pursuing the Ph.D. degree in control science and engineering with Northwestern Polytechnical University, Xi'an, China. His current research interests include aircraft control, UAV, and intelligent control systems.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

Zhenbao Liu (Senior Member, IEEE) received the bachelor's and master's degrees from Northwestern Polytechnical University, Xi'an, China, in 2001 and 2004, respectively, and the Ph.D. degree from the College of Systems and Information Engineering, University of Tsukuba, Tsukuba, Japan, in 2009. He was a Visiting Scholar with Simon Fraser University, Burnaby, BC, Canada, in 2012. He is currently a Professor with Northwestern Polytechnical University. He has authored over 50 papers in major international journals and conferences. His current research interests include flight control, aircraft fault diagnosis, UAV, and intelligent systems.

Junwei Han (Fellow, IEEE) received the Ph.D. degree in pattern recognition and intelligent systems from the School of Automation, Northwestern Polytechnical University, Xi'an, China, in 2003. He is a currently a Professor with Northwestern Polytechnical University. His current research interests include multimedia processing and brain imaging analysis. He is an Associate Editor of IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, *Neurocomputing*, and *Multidimensional Systems and Signal Processing*.

Jinbiao Yuan received the B.S. degree from Hefei University of Technology in 2012, the M.S. degree from Shaanxi University of Science and Technology in 2015, and the Ph.D. degree from Northwestern Polytechnical University in 2025. He is currently a Designer with Qing'an Group Company Ltd. His research interests include UAVs and ground station applications.

Wen Zhao received the M.S. degree in control engineering from Shanghai University, Shanghai, China, in 2014, and the Dr.Eng. degree in modern mechanical engineering from Waseda University, Tokyo, Japan, in 2019. He was a Post-Doctoral Researcher with the Sugano Laboratory, Waseda University, until 2020. He is currently an Associate Professor with the School of Civil Aviation, Northwestern Polytechnical University. He has published several articles and conference papers in the field of robotics and visible light communication. His research interests include robotics, embedded software/hardware design, data processing, sensor technology, control theory and modeling, electronic system design, and visible light communication technology.

Qingqing Dang received the Ph.D. degree from Beihang University, Beijing, China. He is currently an Associate Professor with Northwestern Polytechnical University, Xi'an, China. His current research interests include autonomous unmanned systems, formation control, and intelligent control and simulations.