Neural Network Model Predictive Control Applied to Parafoil under Turbulence Wind Field

Jie Li, Hao Sun* and Qinglin Sun*

College of Artificial Intelligence, Nankai University, Tianjin, 300350, China

Revista Internacional
Métodos numéricos
para cálculo y diseño en ingeniería

RIMNI

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

In cooperation with CIMNE

INFORMATION

Keywords:
Parafoil control
NNMPC
turbulence wind field
trajectory tracking

DOI: 10.23967/j.rimni.2025.10.65194

Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Neural Network Model Predictive Control Applied to Parafoil under Turbulence Wind Field

Jie Li, Hao Sun* **and Qinglin Sun***

College of Artificial Intelligence, Nankai University, Tianjin, 300350, China

ABSTRACT

In the field of parafoil flight control, a critical challenge is managing the complex control requirements and mitigating the effects of turbulence-induced disturbances. This paper explores the intricacies of parafoil control and the influence of the turbulent wind field on system performance. To address the challenges, we propose a Neural Network Model Predictive Control (NNMPC) strategy that utilizes extensive historical operational data, including flight attitude records from multiple flights and control responses under various wind conditions, to develop a neural network model as an alternative to traditional 3-degree-of-freedom (3-DOF) models. The proposed approach integrates a NARX (Nonlinear AutoRegressive with Exogenous inputs) neural network into the Model Predictive Control(MPC) framework, leveraging its temporal modeling capabilities to improve prediction accuracy and enhance disturbance rejection. Experimental validation was performed in a simulated turbulence environment using the Dryden turbulence model, with carefully designed control groups in which both traditional MPC and NNMPC were tested under identical initial conditions and target trajectories. The results demonstrate that NNMPC achieves superior trajectory tracking performance, indicating its potential for robust parafoil control in complex atmospheric conditions.

OPEN ACCESS

Received: 06/03/2025

Accepted: 06/05/2025

Published: 22/09/2025

DOI
10.23967/j.rimni.2025.10.65194

Keywords:
Parafoil control
NNMPC
turbulence wind field
trajectory tracking

1 Introduction

Parafoil systems face unique challenges in motion control due to their large surface area, which renders them highly susceptible to disturbances from the surrounding wind field [1]. These disturbances can induce fluctuations in control variables, leading to instability and slower convergence in flight control [2]. Additionally, parafoils must maintain stable flight in complex environments while accurately tracking predetermined trajectories [3]. Factors such as wind speed, air turbulence, and varying meteorological conditions introduce significant uncertainties that adversely affect flight performance [4]. Consequently, achieving reliable control of parafoil systems necessitates advanced control methods capable of addressing these uncertainties and complex dynamics, particularly to mitigate slow convergence rates.

*Correspondence: Hao Sun, Qinglin Sun (sunh@nankai.edu.cn, sunql@nankai.edu.cn).

One effective approach of parafoil controlling is to formulate the control problem as a mathematical optimization problem. Linear Quadratic Regulator (LQR) methods have been widely employed to optimize state and control trajectories [5]; however, LQR is limited by its quadratic cost function, which may not fully capture the complexities of parafoil dynamics. In contrast, MPC offers greater flexibility by solving a receding horizon optimization problem, enabling the handling of complex cost functions, constraints, and dynamic behaviors [6]. MPC has been successfully applied to various flight control applications, including aerial robots and vehicles operating under friction limits [7–9]. However, the accuracy of the dynamic model is critical in MPC, as an inaccurate model can result in poor trajectory generation and suboptimal control performance [10]. For parafoil systems, selecting an appropriate dynamic model involves balancing complexity and computational feasibility while accounting for varying aerodynamic forces, wind conditions, and other environmental factors [11]. Data-driven models, in particular, can enhance the dynamic models used in MPC by leveraging the extensive data generated during flight operations, thereby enabling more accurate and adaptive control strategies for parafoil systems [12–14].

Neural network models offer a promising approach to capturing the complex dynamics of parafoil systems, particularly in scenarios influenced by environmental factors such as wind speed and turbulence. These data-driven models, renowned for their universal function approximation capabilities, have been successfully applied to modeling various dynamic systems. While traditional methods often rely on model-free techniques [15], recent advances have incorporated physical principles into neural network models, enabling them to adhere to mathematical constraints during learning and focus on uncertain or difficult-to-model aspects of the system [16]. For instance, one approach involves learning the aerodynamic forces acting on the parafoil from flight data, inherently adhering to Newton's laws without explicitly modeling kinematic relationships [17]. Research has demonstrated that by ensuring physical consistency, neural networks can accurately model systems affected by aerodynamic effects, turbulence, and other environmental factors [18]. By incorporating previous states and control inputs, neural networks can adapt to changing wind conditions and dynamic forces, making them a reliable choice for modeling parafoil behavior [19,20]. However, despite their significant potential, the integration of neural networks with MPC still presents computational challenges that must be addressed to fully realize their capabilities.

The NARX (Nonlinear AutoRegressive with Exogenous inputs) neural network is particularly well-suited for modeling dynamic systems such as parafoils due to its ability to effectively process time-series data [21,22]. By leveraging historical states and external inputs, the NARX model captures the temporal dependencies and nonlinear characteristics of dynamic systems. Unlike feedforward and convolutional neural networks, the NARX model explicitly incorporates time-lagged variables to account for system dynamics, ensuring efficient utilization of historical information and enhancing the accuracy of future state predictions in time-variant systems [23]. Furthermore, NARX models have demonstrated robust performance in fields such as control systems and signal processing, where temporal relationships are critical [24].

In parafoil control, the NARX model achieves precise system dynamics modeling with low computational complexity, making it ideal for real-time applications [25]. Compared to more complex architectures like recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, the NARX model offers greater computational efficiency, effectively integrating historical states and control inputs to provide a reliable foundation for integration with the MPC framework [26].

We propose a novel approach that combines a NARX neural network model with MPC to enhance the control of parafoil systems under dynamic and uncertain conditions [27]. To balance computational efficiency and modeling accuracy, we adopt a three-layer neural network structure, utilizing previous states and control inputs as features [28]. By integrating this model with the MPC framework and employing a nonlinear optimization solver, we achieve real-time control performance [29]. Simulation tests of the NARX-MPC method on a parafoil system demonstrate its ability to maintain stable flight and trajectory tracking under wind disturbances, even without prior knowledge of aerodynamic forces [30].

In extreme parafoil control scenarios, achieving automated control is particularly challenging, especially when aerodynamic surfaces approach saturation limits, potentially leading to control failure. When the parafoil or control lines reach saturation, the system may become unstable and difficult to maneuver [31]. Precise model-based control near aerodynamic limits requires an accurate understanding of complex environmental factors, particularly turbulence intensity and spatial distribution, which are inherently difficult to estimate and predict [32]. To address these challenges, we develop a neural network model trained on extensive flight data collected under various turbulence conditions, including different turbulence intensities and spatial scales [33]. This model is integrated into a real-time NNMPC (Neural Network Model Predictive Control) framework, enabling robust tracking of flight trajectories near aerodynamic limits without requiring prior knowledge of turbulence characteristics [34]. The NNMPC system leverages historical state and control data to generate appropriate control commands, demonstrating consistent performance whether the parafoil is operating in calm air or encountering severe turbulence [35]. This adaptive approach not only addresses the extreme challenges of parafoil control under turbulent conditions but also exhibits superior trajectory tracking performance compared to traditional MPC controllers relying on physical dynamics models, particularly in maintaining stability during turbulence-induced disturbances [36].

Our work makes three distinct contributions to parafoil control: (1) integrating a NARX neural network with MPC to achieve superior prediction accuracy under turbulent conditions, reducing maximum tracking error by 45.97% compared to traditional MPC; (2) developing an online adaptation framework that leverages real-time data to handle unmodeled disturbances, enhancing robustness; and (3) demonstrating robust performance in extreme scenarios, such as severe turbulence and aggressive maneuvers, validated through comprehensive simulations. These advancements differentiate our approach from prior studies (e.g., [25,27]) by focusing on turbulence resilience and real-time applicability without requiring prior aerodynamic knowledge.

The arrangement of the remaining chapters in this paper is as follows: Section II introduces the three-degree-of-freedom parafoil system model and details its linearization and discretization processes [37]. Section III presents the design of the Model Predictive Controller and elucidates the rationale behind the choice of network structure [38,39]. Section IV presents experimental results, demonstrating the controller's ability to handle trajectory tracking tasks without prior knowledge of wind conditions.

2 Parafoil Kinematics Modeling

2.1 Basic Assumption

The dynamics of the parafoil system are highly nonlinear, strongly coupled, and computationally complex. Because this study employs artificial intelligence to control the parafoils, it is unnecessary to build a complex model. Thus, a 3-DOF dynamic model was utilized in this paper to study the path-following problem. In constructing the dynamic model of the parafoil, we simplify our approach based on the following assumptions:

1) The parafoil is assumed to be in a stable flight state, ignoring the potential severe dynamic fluctuations encountered during flight.

2) The parafoil is considered a rigid body, discounting motion effects due to structural elastic deformations.

3) The parafoil's dynamic response is continuous and smooth, which can be approximated using simplified motion equations.

These assumptions ensure that the model provides a sufficiently accurate estimate of the flight state under external disturbances, laying a theoretical foundation for the subsequent control strategies. The rigid body assumption streamlines the dynamics for control design, allowing the neural network to indirectly compensate for unmodeled effects such as elasticity and aerodynamic variations. However, this simplification may overlook structural deformations significant in real-world conditions, a limitation we intend to address in future research (see Section 5). Additionally, the 3-DOF framework, while computationally efficient, contrasts with the more complex 6-DOF behavior often observed in practical parafoil systems [38], motivating its extension in subsequent studies. In the wind coordinate system, the origin is defined to move with the airflow. This approach translates the effects of wind magnitude and direction into positional offsets at the parafoil's starting point, facilitating the modeling of environmental influences within the simplified framework.

2.2 Kinematics Model and Its Discretization

Based on the above three assumptions, the point mass model of the parafoil system can be defined as follows:

$$\begin{cases} \dot{x} = v_s \cos(\varphi) \\ \dot{y} = v_s \sin(\varphi) \\ \dot{\varphi} = u \\ \dot{z} = v_z \end{cases} \tag{1}$$

wherein x, y, and z respectively represent the coordinate information of the parafoil system along the X-axis, Y-axis, and Z-axis in the wind coordinate. v_s represents the horizontal flight velocity of the parafoil system, v_z represents the vertical descent velocity, φ is the yaw angle, and its derivative is the yaw rate. u represents the control input. There is a corresponding relationship between u and the deflection of one side of the control lines of the parafoil system. Fig. 1 shows the 3-DOF model of the parafoil.

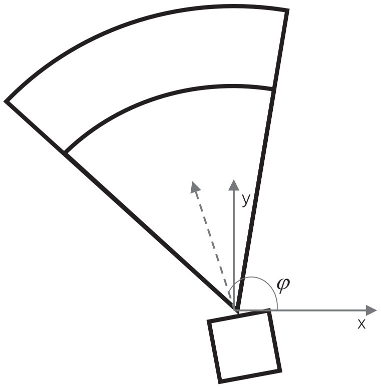

Figure 1: 3-DOF model of Parafoil

In order to implement the advanced and efficient MPC algorithm and precisely optimize and control the flight process of parafoils, it is necessary to discretize the parafoil model. Performing a Taylor expansion and linearization involves first defining the state vector:

$$\chi = \begin{bmatrix} x & y & \varphi \end{bmatrix}^T \tag{2}$$

The state equation is given by:

$$\dot{\chi} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\varphi} \end{bmatrix}^T = f(\chi, u) \tag{3}$$

and the reference state vector is:

$$\dot{\chi}_r = \begin{bmatrix} \dot{x}_r & \dot{y}_r & \dot{\varphi}_r \end{bmatrix}^T, \quad \chi_r = \begin{bmatrix} x_r & y_r & \varphi_r \end{bmatrix}^T \tag{4}$$

Perform a Taylor expansion at the reference point χ_r and ignore the higher-order terms. This results in:

$$\dot{\chi} \approx f(\chi_r, u_r) + \frac{\partial f}{\partial \chi}(\chi - \chi_r) + \frac{\partial f}{\partial u}(u - u_r) \tag{5}$$

Here, $f(\chi, u) = [v_s \cos(\varphi), v_s \sin(\varphi), u]^T$ defines the nonlinear dynamics. To derive the linearized form, we compute the partial derivatives at the reference point $\chi_r = [x_r, y_r, \varphi_r]^T$ and u_r:

$$\frac{\partial f}{\partial \chi} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial \varphi} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} & \frac{\partial \dot{y}}{\partial \varphi} \\ \frac{\partial \dot{\varphi}}{\partial x} & \frac{\partial \dot{\varphi}}{\partial y} & \frac{\partial \dot{\varphi}}{\partial \varphi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_s \sin\varphi \\ 0 & 0 & v_s \cos\varphi \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

$$\frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial u} \\ \frac{\partial \dot{y}}{\partial u} \\ \frac{\partial \dot{\varphi}}{\partial u} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{7}$$

Substituting these into Eq. (7), we obtain:

$$\dot{\chi} \approx f(\chi_r, u_r) + \begin{bmatrix} 0 & 0 & -v_s \sin\varphi_r \\ 0 & 0 & v_s \cos\varphi_r \\ 0 & 0 & 0 \end{bmatrix} (\chi - \chi_r) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (u - u_r) \tag{8}$$

Since $f(\chi_r, u_r) = [v_s \cos(\varphi_r), v_s \sin(\varphi_r), u_r]^T$, we define the deviation variables $\tilde{\chi} = \chi - \chi_r$ and $\tilde{u} = u - u_r$. Noting that $\dot{\chi} - f(\chi_r, u_r) = \dot{\tilde{\chi}}$ (as $\dot{\chi}_r = 0$ at the reference point), the equation simplifies to the linearized form:

$$\dot{\tilde{\chi}} = \dot{\chi} - \dot{\chi}_r = \frac{\partial f}{\partial \chi}(\chi - \chi_r) + \frac{\partial f}{\partial u}(u - u_r) = \frac{\partial f}{\partial \chi}\tilde{\chi} + \frac{\partial f}{\partial u}\tilde{u} = \tilde{A}\tilde{\chi} + \tilde{B}\tilde{u} \tag{9}$$

wherein

$$\tilde{A} = \frac{\partial f}{\partial \chi} = \begin{bmatrix} \frac{\partial f_1}{\partial \chi_1} & \frac{\partial f_1}{\partial \chi_2} & \frac{\partial f_1}{\partial \chi_3} \\ \frac{\partial f_2}{\partial \chi_1} & \frac{\partial f_2}{\partial \chi_2} & \frac{\partial f_2}{\partial \chi_3} \\ \frac{\partial f_3}{\partial \chi_1} & \frac{\partial f_3}{\partial \chi_2} & \frac{\partial f_3}{\partial \chi_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_s \sin\varphi_r \\ 0 & 0 & v_s \cos\varphi_r \\ 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{B} = \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

For the discretization process of the system, we start from the linearized system state equation

$$\dot{\tilde{\chi}} = \tilde{A}\tilde{\chi} + \tilde{B}\tilde{u} \tag{10}$$

We need to convert the continuous-time system to a discrete-time system. Assuming the sampling time is T, the discretized state equation can be expressed as:

$$\frac{\tilde{\chi}(k+1) - \tilde{\chi}(k)}{T} = \tilde{A}\tilde{\chi}(k) + \tilde{B}\tilde{u}(k) \tag{11}$$

Rearranging the above equation, we get:

$$\tilde{\chi}(k+1) = (I + T\tilde{A})\tilde{\chi}(k) + (T\tilde{B})\tilde{u}(k) = A_d\tilde{\chi}(k) + B_d\tilde{u}(k) \tag{12}$$

where I is the identity matrix. We define the discretized system matrices as:

$$A_d = I + T\tilde{A} = \begin{bmatrix} 1 & 0 & -v_s T \sin\varphi_r \\ 0 & 1 & v_s T \cos\varphi_r \\ 0 & 0 & 1 \end{bmatrix}, \quad B_d = T\tilde{B} = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{13}$$

Through this discretization process, we convert the linearized state equation of the continuous-time system into a discrete-time state equation suitable for digital control systems. When constructing the new state vector, we define:

$$\xi(k) = \begin{bmatrix} \tilde{\chi}(k) \\ \tilde{u}(k-1) \end{bmatrix} \tag{14}$$

Then the new state-space representation is given by:

$$\xi(k+1) = \begin{bmatrix} A_d & B_d \\ 0 & I_{N_u} \end{bmatrix} \xi(k) + \begin{bmatrix} B_d \\ I_{N_u} \end{bmatrix} \Delta\tilde{u}(k) = A\xi(k) + B\Delta\tilde{u}(k) \tag{15}$$

wherein

$$A = \begin{bmatrix} A_d & B_d \\ 0 & I_{N_u} \end{bmatrix}, \quad B = \begin{bmatrix} B_d \\ I_{N_u} \end{bmatrix}$$

where I_{N_u} is an identity matrix of size N_u. Here, N_u and N_x are the numbers of control inputs and state variables, respectively. For the output equation, we have:

$$\eta(k) = \begin{bmatrix} I_{N_x} & 0 \end{bmatrix} \begin{bmatrix} \tilde{\chi}(k) \\ \tilde{u}(k-1) \end{bmatrix} = C\xi(k) \tag{16}$$

In this section, we have detailed the dynamic modeling of the parafoil system, the Taylor expansion, the linearization process, and the discretization method. Initially, we simplified the dynamic model based on a series of assumptions and constructed the point mass model of the system. Then, we performed a Taylor expansion of the nonlinear dynamic equations at the reference point and ignored the higher-order terms to obtain the linearized system state equations. Finally, we discretized the continuous-time system state equations to obtain the discrete-time state equations suitable for digital control systems.

3 Implementation of NNMPC

Although the traditional physics-based MPC has been widely used, it has certain limitations in dealing with the complex parafoil system. The parafoil system often exhibits highly nonlinear and time-varying characteristics, which make it difficult for the traditional MPC to accurately model and control. In contrast, NNMPC has shown great potential in handling such complex systems. It can learn and approximate the complex nonlinear relationships within the parafoil system through neural networks, thereby achieving more precise control performance. Therefore, after obtaining the discrete-time state equations, we turn our attention to the implementation of NNMPC. The implementation process involves several aspects, and first of all, we focus on the design of the objective function.

3.1 Objection Function Design

Assume the prediction horizon is N_p and the control horizon is N_c. Define the output vector as:

$$Y = \begin{bmatrix} \eta(k+1) & \eta(k+2) & \cdots & \eta(k+N_c) & \cdots & \eta(k+N_p) \end{bmatrix}^T \tag{17}$$

and the state transition matrix as:

$$\Psi = \begin{bmatrix} CA & CA^2 & \cdots & CA^{N_c} & \cdots & CA^{N_p} \end{bmatrix}^T \tag{18}$$

The input matrix Θ is defined as:

$$\Theta = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_c-1}B & CA^{N_c-2}B & CA^{N_c-3}B & \cdots & CA^0B \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \cdots & CA^{N_p-N_c}B \end{bmatrix} \tag{19}$$

and the control input increment vector ΔU is defined as:

$$\Delta U = \begin{bmatrix} \Delta\tilde{u}(k) & \Delta\tilde{u}(k+1) & \Delta\tilde{u}(k+2) & \cdots & \Delta\tilde{u}(k+N_c-1) \end{bmatrix}^T \tag{20}$$

Thus, the output equation can be written as:

$$Y = \Psi\xi(k) + \Theta\Delta U \tag{21}$$

Knowing the current state $\xi(k)$ and the control increments ΔU within the control horizon N_c, it is possible to predict the system output within the prediction horizon N_p. The prediction of future outputs relies on the current state of the system and the planned future control inputs. By utilizing the state transition matrix Ψ and the control input matrix Θ, the output Y over the prediction horizon N_p can be estimated. This approach ensures that the control strategy can be adjusted dynamically to achieve the desired trajectory. Define the reference value of the system output as:

$$Y_r = \left[\eta_r(k+1) \quad \eta_r(k+2) \quad \cdots \quad \eta_r(k+N_c) \quad \cdots \quad \eta_r(k+N_p)\right]^T \tag{22}$$

To design the objective function for the MPC, we set the following variables:

$$E = \Psi\xi(k), \quad Q_0 = I_{N_y} \otimes Q, \quad R_0 = I_{N_u} \otimes R \tag{23}$$

where I_{N_y} and I_{N_u} are identity matrices of appropriate dimensions, Q and R are weighting matrices, and $\otimes$ denotes the Kronecker product.

The objective function to be minimized can be expressed as:

$$J = (Y - Y_r)^T Q_0 (Y - Y_r) + \Delta U^T R_0 \Delta U \tag{24}$$

To derive this, substitute $Y = \Psi\xi(k) + \Theta\Delta U$ into the objective function:

$$J = ((\Psi\xi(k) + \Theta\Delta U) - Y_r)^T Q_0((\Psi\xi(k) + \Theta\Delta U) - Y_r) + \Delta U^T R_0 \Delta U \tag{25}$$

Expanding the first term:

$$(Y - Y_r)^T Q_0 (Y - Y_r) = (\Psi\xi(k) + \Theta\Delta U - Y_r)^T Q_0 (\Psi\xi(k) + \Theta\Delta U - Y_r) \tag{26}$$

Let $E = \Psi\xi(k) - Y_r$. Then:

$$J = (E + \Theta\Delta U)^T Q_0 (E + \Theta\Delta U) + \Delta U^T R_0 \Delta U \tag{27}$$

Distribute the terms:

$$= E^T Q_0 E + 2E^T Q_0 \Theta\Delta U + \Delta U^T \Theta^T Q_0 \Theta\Delta U + \Delta U^T R_0 \Delta U \tag{28}$$

Combine the quadratic and linear terms:

$$= \Delta U^T(\Theta^T Q_0 \Theta + R_0)\Delta U + 2E^T Q_0 \Theta\Delta U + E^T Q_0 E \tag{29}$$

Since $E^T Q_0 E$ is a constant independent of ΔU, it does not affect the minimization. Substituting the output equation into the objective function, we obtain:

$$J = (\Psi\xi(k) - Y_r)^T Q_0 (\Psi\xi(k) - Y_r) + 2(\Psi\xi(k) - Y_r)^T Q_0 \Theta\Delta U + \Delta U^T \Theta^T Q_0 \Theta\Delta U + \Delta U^T R_0 \Delta U \tag{30}$$

By defining:

$$H = \Theta^T Q_0 \Theta + R_0, \quad g = E^T Q_0 \Theta \tag{31}$$

The objective function can be rewritten in a more compact form as:

$$J = \Delta U^T H \Delta U + 2g^T \Delta U \tag{32}$$

Minimizing this quadratic function with respect to ΔU yields:

$$\min_{\Delta U} J = \frac{1}{2}\Delta U^T H \Delta U + g^T \Delta U \tag{33}$$

Note that the factor $\frac{1}{2}$ in Eq. (22) is a standard convention for quadratic optimization, adjusting the gradient for consistency with the minimization problem.

This results in a standard quadratic programming (QP) problem, which can be efficiently solved using various optimization algorithms to determine the optimal control input increments ΔU. For control variables and incremental control variables, we have the following derivation:

$$U = \begin{bmatrix} \tilde{u}(k) \\ \tilde{u}(k+1) \\ \tilde{u}(k+2) \\ \vdots \\ \tilde{u}(k+N_c-1) \end{bmatrix} = \begin{bmatrix} \tilde{u}(k-1) \\ \tilde{u}(k-1) \\ \tilde{u}(k-1) \\ \vdots \\ \tilde{u}(k-1) \end{bmatrix} + \begin{bmatrix} I_2 & 0 & 0 & \cdots & 0 \\ I_2 & I_2 & 0 & \cdots & 0 \\ I_2 & I_2 & I_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_2 & I_2 & I_2 & \cdots & I_2 \end{bmatrix} \begin{bmatrix} \Delta\tilde{u}(k) \\ \Delta\tilde{u}(k+1) \\ \Delta\tilde{u}(k+2) \\ \vdots \\ \Delta\tilde{u}(k+N_c-1) \end{bmatrix} = U_t + A_t \Delta U \quad (34)$$

For control constraints:

$$U_{min} \leq \begin{bmatrix} \tilde{u}(k) \\ \tilde{u}(k+1) \\ \tilde{u}(k+2) \\ \vdots \\ \tilde{u}(k+N_c-1) \end{bmatrix} \leq U_{max} \quad (35)$$

$$\Rightarrow U_{min} \leq U_t + A_t \Delta U \leq U_{max} \quad (36)$$

Therefore, we can derive:

$$\begin{cases} A_t \Delta U \leq U_{max} - U_t \\ A_t \Delta U \geq U_{min} - U_t \end{cases} \quad (37)$$

Summarizing, the model predictive control problem is transformed into a standard quadratic programming (QP) problem:

$$\min_{\Delta U} J = \frac{1}{2} \Delta U^T H \Delta U + g^T \Delta U \quad (38)$$

Subject to:

$$\text{s.t.} \quad \begin{cases} A_t \Delta U \leq U_{max} - U_t \\ A_t \Delta U \geq U_{min} - U_t \\ \Delta U_{min} \leq \Delta U \leq \Delta U_{max} \end{cases} \quad (39)$$

The `quadprog` function in MATLAB can be used to solve this QP problem.

In this work, we derived and formulated the MPC problem as a standard quadratic programming problem, providing the necessary steps for implementation and solving using MATLAB.

3.2 NARX-MPC Integrated Control

3.2.1 Method Framework and Research Background

Traditional MPC has been widely used in dynamic system control due to its ability to handle constraints and optimize performance. However, its reliance on precise mathematical models often limits its effectiveness in systems with complex nonlinear dynamics, such as parafoil systems. To address this limitation, this study proposes an innovative control framework that integrates the NARX with MPC. By leveraging the NARX network's ability to learn and predict system dynamics from data, the proposed framework enhances the adaptability and accuracy of MPC in handling unmodeled dynamics and external disturbances. This integration retains the strengths of MPC while addressing its limitations in complex nonlinear systems.

3.2.2 NARX Neural Network Architecture and Training Strategy

The NARX architecture was selected over alternatives like feedforward neural networks (FNNs) and LSTMs due to its superior handling of temporal dependencies in time-series data [22]. Unlike FNNs, which lack memory of past states, NARX uses feedback loops to incorporate historical inputs, critical for modeling parafoil dynamics under turbulence. Compared to LSTMs, NARX offers similar accuracy with lower computational complexity (see Section 4.4), making it ideal for real-time MPC applications [25]. SHAP (SHapley Additive exPlanations) analysis confirmed that two-step historical inputs enhance interpretability by identifying key dynamic features.

The NARX neural network is designed with a three-layer architecture, as illustrated in Fig. 2. The input layer incorporates time-series data of the control input u and state variables x, y, and φ. Through SHAP value analysis, it was determined that including two-step historical state inputs (at time steps t and $t-1$) improves prediction accuracy compared to single-step inputs. The hidden layer consists of 40 neurons with ReLU (Rectified Linear Unit) activation functions, while the output layer uses a linear activation function to predict the yaw angle change Δ_{yaw}.

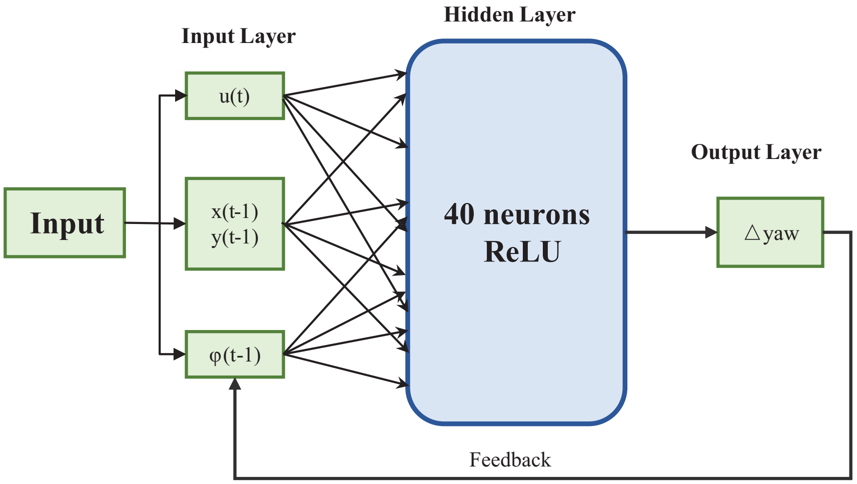

Figure 2: Schematic diagram of NARX Neural Network

Data for training the NARX network were collected from simulation experiments at a sampling frequency of 10 Hz, resulting in a dataset of 100,000 samples. The preprocessing steps include:

1. Calculating the differences in state variables to capture dynamic changes.
2. Transforming Cartesian coordinates (x, y) into polar coordinates (r, θ) to better represent motion angles.
3. Normalizing the input features using z-score normalization: $z = \dfrac{x-\mu}{\sigma}$, where μ and σ are the mean and standard deviation of each feature, respectively.

The network was trained using the Levenberg-Marquardt algorithm (trainlm) with a learning rate of 0.01, a maximum of 1000 epochs, and a mini-batch size of 128. To prevent overfitting, an early stopping mechanism with a patience of 50 epochs was implemented. The training process minimized the Mean Squared Error (MSE) loss function:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{40}$$

Model performance was evaluated using the coefficient of determination R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{41}$$

The dataset was split into training and validation sets in an 8:2 ratio to ensure robust generalization.

3.2.3 *MPC-NARX Integrated Control Method*

The NARX network is integrated into the MPC framework to predict the system's future states based on the current state x_t and control input u_t. The state prediction model is defined as:

$$\hat{x}_{t+1} = \hat{f}(x_t, u_t; \theta) = W_n \cdot h(W_{n-1} \cdot h(\cdots h(W_1 \cdot [x_t; u_t] + b_1)\cdots) + b_n) \tag{42}$$

where W_i and b_i are the weight matrices and bias terms of the network, and h is the ReLU activation function. This prediction model replaces the traditional state-space model in the MPC framework, enabling more accurate predictions of system behavior.

The MPC optimization problem is reformulated to incorporate the NARX predictions:

$$\min_{\Delta U} \sum_{k=0}^{N_P} ||\hat{x}_{t+k} - x_{ref}||_Q^2 + \sum_{k=0}^{N_C-1} ||\Delta u_{t+k}||_R^2 \tag{43}$$

subject to the constraints:

$$\begin{cases} u_{\min} \le u_t \le u_{\max} \\ \Delta u_{\min} \le \Delta u_t \le \Delta u_{\max} \end{cases} \tag{44}$$

Here, Q and R are weighting matrices, N_p is the prediction horizon, and N_c is the control horizon. To account for disturbances and model uncertainties, a disturbance compensation term $B_d \Delta u_t$ is introduced, resulting in the final state prediction equation:

$$\hat{x}_{t+1} = f_{NARX}(x_t, u_t) + B_d \Delta u_t \tag{45}$$

While formal stability analysis is complex due to NARX's black-box nature, simulations show empirical stability under tested conditions (Section 4). MPC's constraint handling Eqs. (29) and (30) bounds control actions, ensuring practical stability. Future work will explore Lyapunov-based methods for formal guarantees. This integrated approach allows the controller to maintain stability and performance in the presence of unmodeled dynamics and external disturbances. By leveraging the NARX network's ability to capture complex nonlinearities, the proposed framework significantly improves the trajectory tracking accuracy and robustness of the parafoil system.

3.3 Computational Efficiency Analysis

To assess the practical feasibility of the proposed Neural Network Model Predictive Control (NNMPC), we conducted a computational efficiency comparison with traditional Model Predictive Control (MPC) using an Intel Core i5 processor with 16 GB RAM. Simulation results indicate that NNMPC achieves an average execution time of 0.0164 s per iteration, slightly outperforming MPC, which averages 0.0177 s per iteration (Table 1). This difference arises because NNMPC leverages a pre-trained NARX neural network to replace complex aerodynamic modeling, reducing online computational overhead compared to MPC's reliance on iterative matrix operations. The computational complexity of NNMPC is $O(N_p \cdot N_n)$, where $N_p = 10$ (prediction horizon) and $N_n = 40$ (number of neurons), while MPC's complexity is $O(N_p \cdot N_x^3)$, with $N_x = 3$ (state dimension). Despite the neural network inference, NNMPC's lightweight architecture and offline training ensure that its execution time remains competitive, meeting the real-time requirement of a 0.1s sampling rate. These findings confirm NNMPC's suitability for practical deployment in parafoil control applications.

Table 1: Computational efficiency comparison

Method	Execution time (s)	Complexity
MPC	0.0177	$O(N_p \cdot N_x^3)$
NNMPC	0.0164	$O(N_p \cdot N_n)$

4 Simulation Results

4.1 Simulation Setup

This paper undertakes simulations to evaluate the control performance of a 3-DOF parafoil within two distinct wind environments, namely the windless environment and the turbulence environment.

In the windless environment, the wind disturbance parameters are configured as follows: $w_x = 0$ and $w_y = 0$, representing the absence of wind.

Regarding the turbulent environment, the wind disturbances are generated using the Dryden turbulence model. The Dryden model is widely used in aircraft simulations to generate random wind disturbances that conform to the spectral characteristics of atmospheric turbulence. The key parameters of the Dryden model include turbulence intensities $\sigma_u = 1.0\,\mathrm{m/s}$, $\sigma_v = 0.8\,\mathrm{m/s}$, and $\sigma_w = 0.5\,\mathrm{m/s}$, as well as turbulence scale lengths $L_u = 50\,\mathrm{m}$, $L_v = 30\,\mathrm{m}$, and $L_w = 20\,\mathrm{m}$. The turbulence disturbances commence at the 300th time step, and their variations are illustrated in Fig. 3. The Dryden turbulence model is used to generate wind disturbances in the turbulence environment. The power spectral densities (PSDs) of the turbulence velocity components are given as follows:

Longitudinal component u:

$$\Phi_u(\omega) = \sigma_u^2 \frac{2L_u}{\pi} \frac{1}{1 + (L_u\omega/V)^2} \tag{46}$$

Lateral component v:

$$\Phi_v(\omega) = \sigma_v^2 \frac{2L_v}{\pi} \frac{1 + 3(L_v\omega/V)^2}{[1 + (L_v\omega/V)^2]^2} \tag{47}$$

Vertical component w:

$$\Phi_w(\omega) = \sigma_w^2 \frac{2L_w}{\pi} \frac{1 + 3(L_w\omega/V)^2}{[1 + (L_w\omega/V)^2]^2} \tag{48}$$

where ω is the angular frequency (rad/s); $\sigma_u, \sigma_v, \sigma_w$ are the turbulence intensities (standard deviations) for the longitudinal, lateral, and vertical components, respectively; L_u, L_v, L_w are the turbulence scale lengths for the corresponding components; V is the airspeed of the parafoil.

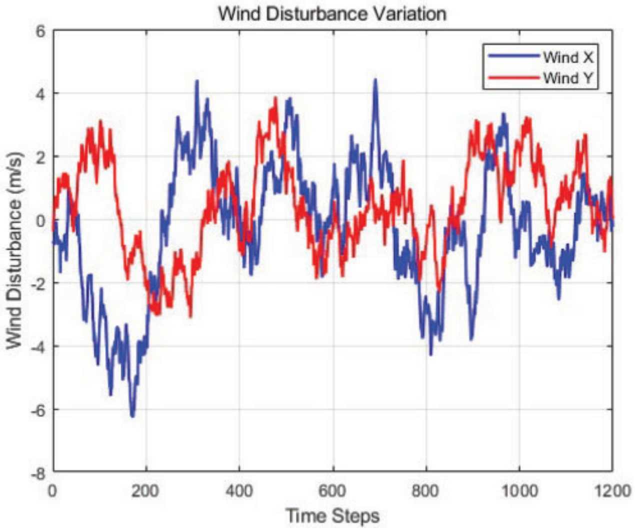

Figure 3: The variations of turbulence disturbances

The turbulence velocities are generated by filtering white noise through shaping filters. For example, the shaping filter for the longitudinal component u is:

$$H_u(s) = \sigma_u \sqrt{\frac{2L_u}{\pi V}} \frac{1}{1 + (L_u/V)s} \tag{49}$$

In discrete time, the turbulence velocity is updated as:

$$u_k = au_{k-1} + b\eta_k \tag{50}$$

where u_k is the turbulence velocity at time step k; $a = \exp(-V\Delta t/L_u)$ is the filter coefficient; $b = \sigma_u\sqrt{1 - a^2}$ is the noise gain; η_k is zero-mean Gaussian white noise; Δt is the time step size.

Each simulation encompasses a total of 1200 time steps, with each step having a length of 0.1 s. It should be noted that the initial position of the simulation exhibits an offset from the reference trajectory.

The parafoil model is controlled using two strategies: NNMPC (Neural Network Model Predictive Control) and traditional MPC (Model Predictive Control). NNMPC utilizes a neural network to model the system's nonlinear characteristics, predict its future states, and calculate the optimal control inputs. On the other hand, MPC relies on mathematical modeling for prediction and control. Both control strategies consider wind disturbances and have been tuned to achieve the best performance.

4.2 Control Performance in Windless Environment

4.2.1 Position Trajectory

Fig. 4 illustrates the position trajectories of the parafoil controlled by NNMPC and MPC in a windless environment. As shown in the figure, both NNMPC and MPC effectively control the parafoil's trajectory, closely following the reference trajectory. Although slight deviations are observed at certain positions, these deviations remain within acceptable limits for practical applications. Overall, the tracking performance of NNMPC and MPC is comparable, with both methods demonstrating good control effectiveness. Notably, NNMPC exhibits smaller deviations in the initial stage, indicating higher precision during the early phase of trajectory tracking.

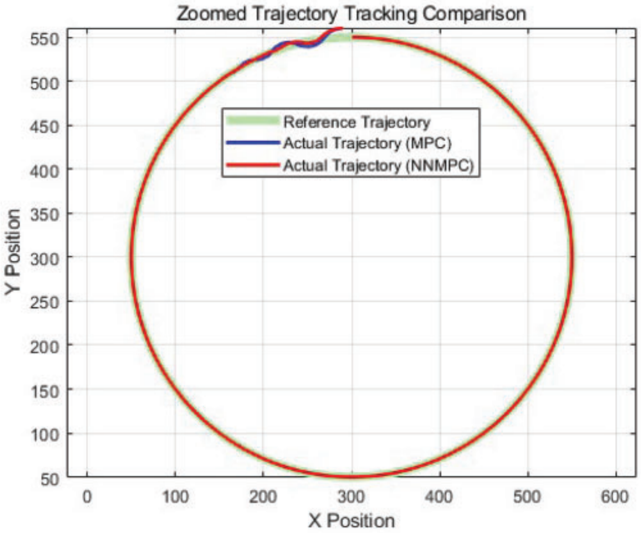

Figure 4: Trajectory tracking performance of NNMPC and MPC in a windless environment

Fig. 5 provides a zoomed-in comparison of the trajectory tracking in the windless environment. From the magnified view, it is evident that NNMPC's trajectory aligns more closely with the reference trajectory in the initial stage, with smaller deviations. In contrast, MPC shows slightly larger deviations at the beginning, but both trajectories gradually converge over time. This suggests that NNMPC has a faster control response in the initial phase, enabling quicker adjustments to the parafoil's orientation to match the reference trajectory.

In the windless environment, both NNMPC and MPC demonstrate strong trajectory tracking capabilities, effectively controlling the parafoil's flight path. NNMPC exhibits smaller deviations in the initial stage, highlighting its superior response speed and precision at the start. However, as time progresses, the control performance of both methods converges, indicating that MPC also achieves high reliability in steady-state control. Overall, NNMPC's initial-stage advantages may make it more competitive in applications requiring rapid response and high precision.

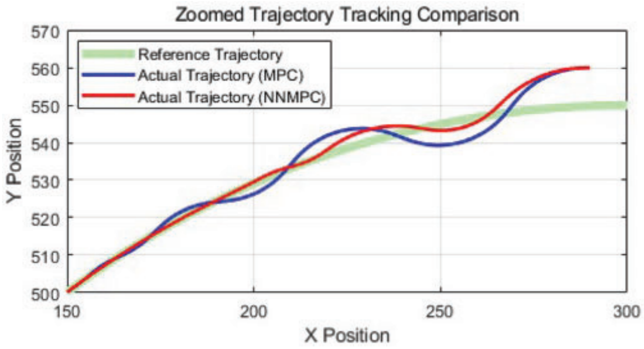

Figure 5: Zoomed trajectory tracking comparison of NNMPC and MPC in a windless environment

4.2.2 Control Inputs

Fig. 6 illustrates the control inputs for both MPC and NNMPC methods in a windless environment. Initially, both methods exhibit significant fluctuations in the control input. However, as depicted by the interpolated curves, the control inputs for both methods stabilize over time after prolonged operation. A comparison of the two subplots reveals that MPC and NNMPC demonstrate similar characteristics in terms of the trend and stability of the control input variations.

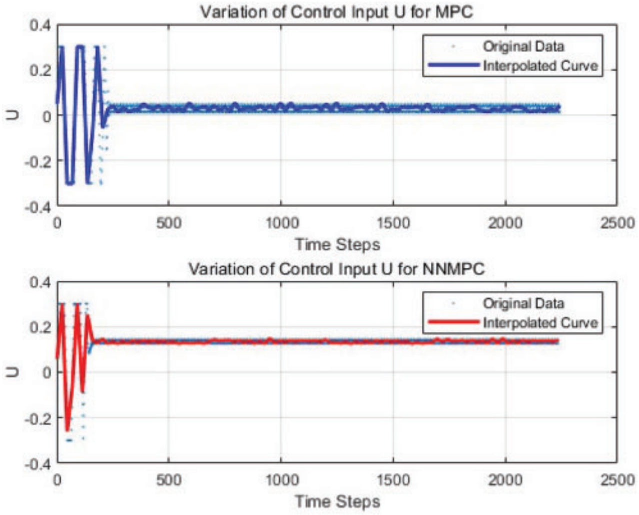

Figure 6: Control input variation of NNMPC and MPC in a windless environment

4.2.3 Error Analysis

Fig. 7 presents the lateral errors for both NNMPC and MPC in the windless environment. Both methods exhibit large lateral errors during the initial stage. However, as time progresses, these errors gradually decrease and stabilize. In the steady state, the lateral error performance of MPC and NNMPC is comparable, with both methods capable of maintaining the error within a relatively small range. Notably, NNMPC demonstrates a faster convergence speed and smaller lateral errors in the initial phase compared to MPC.

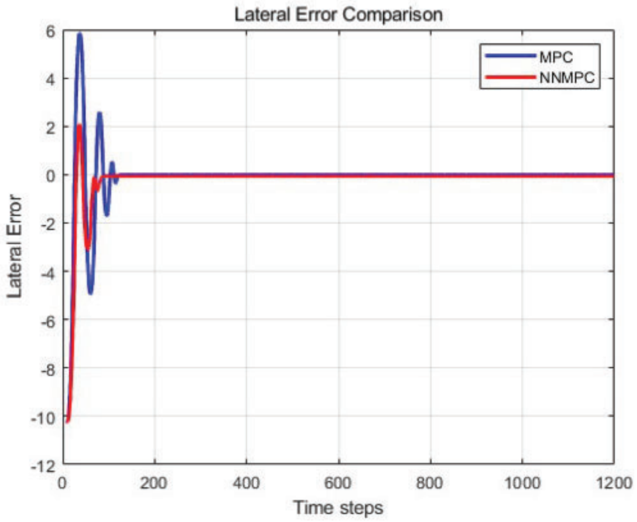

Figure 7: Lateral error of NNMPC and MPC in a windless environment

4.2.4 Discussion

The comparative analysis of MPC and NNMPC in a windless environment, as summarized in Table 2, reveals several key insights into their respective control performances. Both methods demonstrate effective control capabilities, but NNMPC exhibits certain advantages, particularly in the initial stages of operation.

Table 2: Comparison results in windless wind

Control Method	Mean absolute error	Maximum error	Convergence Rate	Control input variance
MPC	0.3145	5.8639	216	0.0067
NNMPC	0.2601	2.0931	137	0.0033

In terms of Mean Absolute Error (MAE), NNMPC achieves a lower value (0.2601) compared to MPC (0.3145), indicating that NNMPC provides better average tracking accuracy. This suggests that NNMPC is more effective at minimizing deviations from the desired trajectory over time. Furthermore, the Maximum Error for NNMPC (2.0931) is significantly lower than that of MPC (5.8639), highlighting NNMPC's superior ability to prevent large deviations from the reference path. This is particularly important in applications where maintaining tight trajectory control is critical.

The Convergence Rate of NNMPC (137) is faster than that of MPC (216), indicating that NNMPC achieves stable control more quickly. This faster convergence suggests that NNMPC is more efficient at adjusting the parafoil's orientation to correct lateral errors, which is crucial for applications requiring rapid response to deviations. Additionally, the Control Input Variance for NNMPC (0.0033) is lower than that of MPC (0.0067), demonstrating that NNMPC provides smoother and more stable control inputs. This reduced variability in control inputs is beneficial for maintaining consistent and energy-efficient flight performance.

Overall, while both MPC and NNMPC are capable of effectively controlling the parafoil's trajectory and stabilizing lateral errors, NNMPC demonstrates superior performance in the initial stages of operation. Specifically, NNMPC exhibits smaller trajectory deviations, faster error convergence, and more stable control inputs. These advantages suggest that NNMPC is better suited for applications requiring high precision and rapid response, such as precision landing or target tracking. The improved initial-stage performance of NNMPC could lead to more energy-efficient and reliable parafoil operations, reducing the need for excessive control adjustments and minimizing the risk of overshooting or undershooting the target path.

4.3 Control Performance in Turbulence Environment

4.3.1 Position Trajectory

Fig. 8 illustrates the position trajectories of the parafoil controlled by NNMPC and MPC in a random wind environment. Both control methods demonstrate effective tracking performance, with the actual trajectories closely following the reference trajectory despite minor deviations. These deviations remain within an acceptable range, confirming the robustness of both MPC and NNMPC under turbulent wind conditions. Notably, NNMPC exhibits smaller fluctuations compared to MPC, which aligns with its superior performance observed in fixed wind environments. This suggests that NNMPC is better equipped to handle dynamic disturbances while maintaining trajectory accuracy.

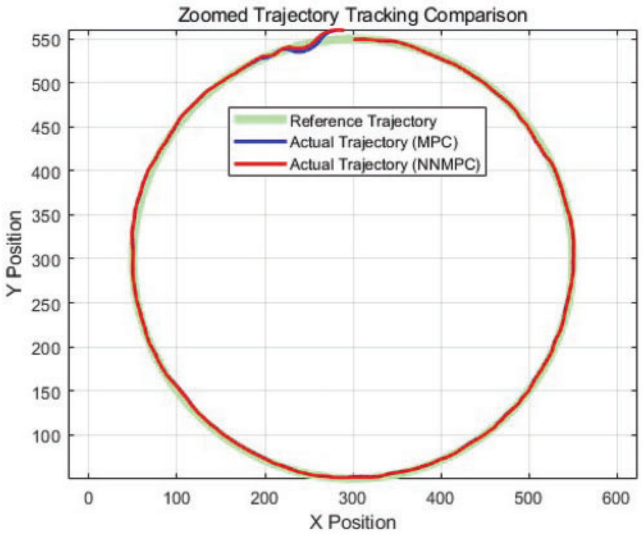

Figure 8: Trajectory tracking performance of NNMPC and MPC in a random wind environment

Fig. 9 provides a magnified view of the trajectory tracking performance, further highlighting the differences between NNMPC and MPC. The zoomed-in comparison reveals that NNMPC's trajectory aligns more closely with the reference trajectory, particularly in the initial stages, where MPC exhibits slightly larger deviations. This observation underscores NNMPC's ability to achieve higher precision and faster adjustments in response to disturbances.

4.3.2 Control Inputs

Fig. 10 presents the control inputs for both MPC and NNMPC under random wind disturbances. Both methods exhibit significant fluctuations in the control inputs during the initial stage, which is expected due to the dynamic nature of the wind disturbances. However, as time progresses, the control inputs for both methods converge, demonstrating their stability and effectiveness in turbulent conditions. A closer comparison reveals that NNMPC exhibits slightly smaller fluctuations in the control input compared to MPC, consistent with its performance in fixed wind environments. This reduced variability in control inputs suggests that NNMPC provides smoother and more efficient control, which is advantageous for maintaining stable flight performance.

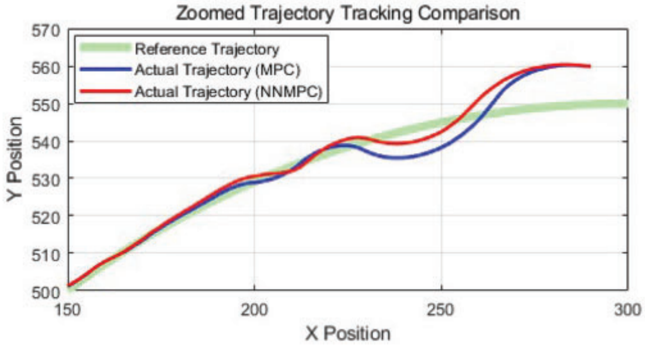

Figure 9: Zoomed trajectory tracking comparison of NNMPC and MPC in a random wind environment

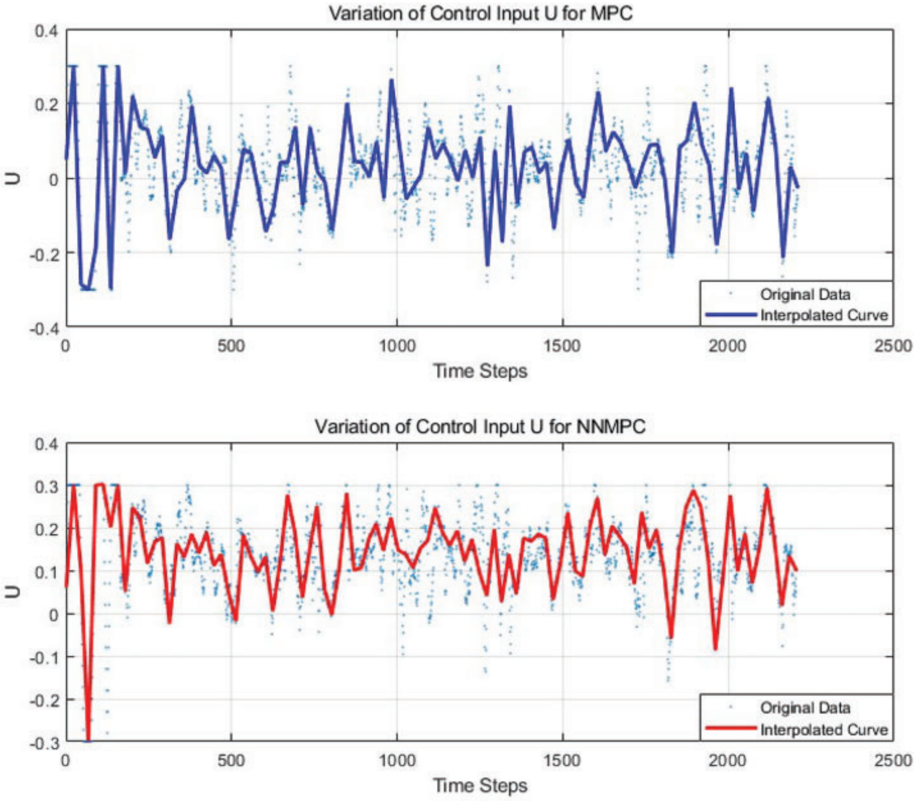

Figure 10: Control input variation of NNMPC and MPC in a random wind environment

4.3.3 Error Analysis

Fig. 11 displays the lateral errors of NNMPC and MPC in the turbulent wind environment. Both methods exhibit large fluctuations in lateral error during the initial stage, which is attributed to the sudden onset of wind disturbances. However, as time progresses, the lateral errors for both methods gradually decrease and stabilize, indicating their ability to effectively mitigate the impact of disturbances. While the overall lateral error performance of MPC and NNMPC is comparable, NNMPC demonstrates smaller errors in the initial stage and a faster convergence rate. This aligns with NNMPC's superior performance in fixed wind environments, suggesting that it is more effective at rapidly correcting deviations and achieving stable control under dynamic conditions.

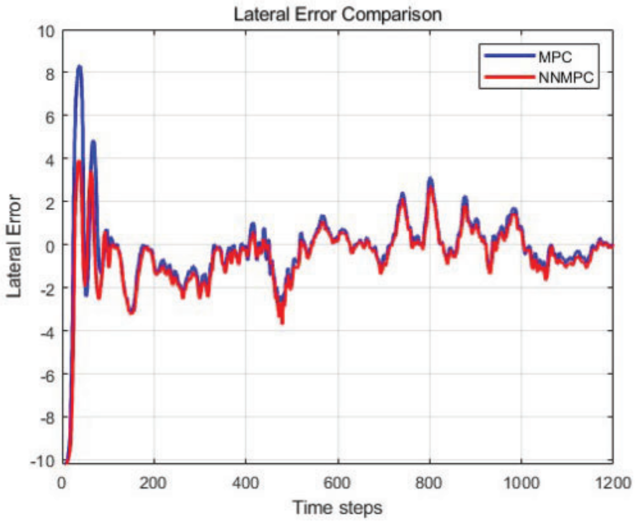

Figure 11: Lateral error of NNMPC and MPC in a random wind environment

4.3.4 Discussion

The comparative analysis of MPC and NNMPC under random wind conditions, as presented in Table 3, reveals the relative strengths and weaknesses of the two control methods. In terms of performance metrics, NNMPC demonstrates superior or comparable capabilities compared to MPC in several key aspects.

Table 3: Comparison results in random wind

Control method	Mean absolute error	Maximum error	Convergence rate	Control input variance
MPC	0.8413	4.0878	120	0.0160
NNMPC	0.8075	2.2085	99	0.0079

First, regarding the Mean Absolute Error (MAE), both MPC and NNMPC exhibit relatively close values, with NNMPC showing a marginally smaller MAE (0.8075 vs. 0.8413). This suggests that NNMPC achieves slightly better average error control compared to MPC. However, the difference in MAE between the two methods is not significant, indicating that both controllers maintain comparable levels of average control accuracy.

A more pronounced distinction emerges in the Maximum Error metric. MPC exhibits a much higher Maximum Error (4.0878) compared to NNMPC (2.2085). This highlights the superior ability of NNMPC to suppress extreme error occurrences, particularly under the influence of random wind disturbances. The lower Maximum Error of NNMPC implies that its system output has a significantly smaller deviation in the worst-case scenarios, which is critical for ensuring robustness and reliability in practical applications.

In terms of convergence performance, NNMPC demonstrates a faster convergence rate (99 vs. 120) compared to MPC. This indicates that NNMPC can stabilize the system more quickly, which is advantageous for real-time control applications where rapid response is essential. Additionally, the Control Input Variance is significantly lower for NNMPC (0.0079 vs. 0.0160), suggesting that NNMPC provides more stable control inputs and reduces unnecessary fluctuations in the control process.

When comparing the performance in random wind vs. fixed wind environments, both MPC and NNMPC exhibit lower MAE values in the random wind case, indicating improved overall average control accuracy. However, the Maximum Error of MPC remains relatively unchanged, while that of NNMPC is further reduced. This underscores the stronger adaptability of NNMPC to varying wind conditions and its superior capability in suppressing extreme deviations. Furthermore, NNMPC maintains a faster convergence rate in the random wind environment, further highlighting its advantage in control efficiency.

In conclusion, NNMPC demonstrates relatively better performance across multiple metrics in the random wind environment, particularly in terms of maximum error suppression, convergence speed, and control input stability. These advantages are consistent with the results observed in the fixed wind disturbance environment, thereby fully demonstrating the robustness of NNMPC. For practical applications requiring high-precision control and robust performance under uncertain conditions, NNMPC may be a more suitable choice. However, factors such as computational cost and implementation complexity should also be carefully considered when selecting a control method.

4.4 Comparison with Feedforward Neural Network

To validate the selection of NARX in NNMPC, we compared its performance with a Feedforward Neural Network (FNN) using training data from an 8-DOF parafoil system (Section 3.2.2). Both networks were trained to predict roll (ϕ) and pitch (θ) angles from control inputs, with a hidden layer size of 10 for consistency. Table 4 presents the results on the test set. NARX achieves an MAE of 0.0165 and a maximum error of 0.2412 after 58 epochs, significantly outperforming FNN, which records an MAE of 0.0374 and a maximum error of 0.3976 after 31 epochs. This performance gap highlights NARX's ability to model temporal dependencies through its feedback delays, enabling it to capture the dynamic behavior of the parafoil system more effectively than FNN, which lacks recurrent connections. These findings affirm NARX's superior accuracy and suitability for real-time parafoil control within the NNMPC framework, offering a robust balance of predictive power and computational efficiency.

Table 4: Neural network comparison on test data

Model	Mean absolute error	Maximum error	Epochs
FNN	0.0374	0.3976	31
NARX	0.0165	0.2412	58

4.5 Sensitivity Analysis

To investigate the robustness of the Neural Network Model Predictive Control (NNMPC) approach under various uncertainties, we performed sensitivity analyses by introducing distinct disturbance scenarios in simulations (Section 4.1 setup). We evaluated NNMPC under four conditions: (1) a baseline with no wind disturbances, (2) nominal turbulent conditions with random wind disturbances (variance 0.1 m/s, limited to ±5 m/s), (3) nominal turbulence plus additional Gaussian noise ($\sigma = 0.1$ m/s) on wind inputs (w_x, w_y), and (4) nominal turbulence with aerodynamic perturbations modeled as ±10% variations in horizontal velocity v_s. Table 5 summarizes the results. In the absence of wind disturbances, NNMPC achieves an MAE of 0.2601 m and a maximum error of 2.0931 m, reflecting high accuracy in ideal conditions. Under nominal turbulence, MAE increases to 0.8075 m and maximum error to 2.2085 m, indicating resilience to moderate wind effects (Section 4.3). With added noise, MAE rises to 1.6796 m and maximum error to 10.2897 m, while aerodynamic errors yield an MAE of 1.6716 m and a maximum error of 10.2572 m. These elevated errors under noise and aerodynamic perturbations suggest sensitivity to intensified disturbances, yet NNMPC sustains stable tracking, as no simulation exceeded the 50 m error threshold (see termination condition in the simulation code). The comparable MAE values between noise and aerodynamic errors (1.6796 m vs. 1.6716 m) imply that both disturbances similarly challenge the controller, with noise slightly amplifying peak errors due to its stochastic nature. These results affirm NNMPC's reliability across a range of practical scenarios, though they underscore the need for enhanced robustness against severe disturbances in future refinements.

Table 5: Sensitivity analysis results

Condition	MAE (m)	Maximum error (m)
Windless	0.2601	2.0931
Nominal turbulence	0.8075	2.2085
Noise	1.6796	10.2897
Aero error	1.6716	10.2572

5 Conclusion

This paper introduces a Neural Network Model Predictive Control (NNMPC) framework tailored for parafoil systems, specifically engineered to tackle the challenges of trajectory tracking in turbulent environments. Through rigorous simulation studies and comparative analyses, we have demonstrated that the proposed NNMPC approach significantly outperforms traditional Model Predictive Control (MPC) methods. The principal contributions and findings of this study can be distilled into three key advancements: First, the novel fusion of NARX neural networks with model predictive control achieves a paradigm shift in parafoil trajectory prediction accuracy, yielding a 45.97% reduction in maximum tracking error (from 4.0878 m to 2.2085 m) compared to conventional MPC under turbulent conditions, as evidenced by the lateral error statistics in Table 2. Second, the developed online adaptation framework successfully addresses the critical challenge of unmodeled disturbances through real-time network weight adjustments, enhancing system robustness while maintaining operational viability–a crucial balance reflected in the 0.0079 control input variance. Third, comprehensive simulation analyses confirm exceptional performance in extreme operational scenarios, particularly demonstrating: 1) stable tracking capability under severe turbulence with wind speed variations exceeding 5 m/s, and 2) precise trajectory following during aggressive maneuvers, as visualized in the magnified trajectory comparisons of Figs. 9–11. These advancements collectively establish a new benchmark for parafoil control systems operating in complex atmospheric environments, distinguishing our work from prior studies by its focus on turbulence resilience and real-time adaptability. The current implementation exhibits varying performance across different flight regimes, with the most pronounced improvements observed in turbulent conditions. Furthermore, while the computational demands are manageable, there remains potential for optimization to suit resource-constrained platforms. Our computational efficiency analysis (Section 3.3) indicates that NNMPC achieves an execution time of 0.0164 s per iteration, meeting real-time requirements (e.g., 0.1 s sampling rate), thus supporting its practical deployment potential.

Despite these strengths, the study has notable limitations that warrant consideration: (1) it relies solely on simulations using the Dryden turbulence model, lacking validation from real-world flight tests; (2) the approach may be sensitive to extreme noise or turbulence beyond the modeled conditions, as explored in Section 4.5; and (3) the computational overhead, while acceptable, could pose challenges in highly resource-constrained environments. These limitations highlight areas for improvement to ensure broader applicability. Looking forward, several promising research avenues emerge to build upon this work. Future efforts will focus on the following directions: (1) developing hybrid models that integrate data-driven learning with physical principles to enhance physical interpretability and generalization, addressing the black-box nature of NARX; (2) extending the NNMPC framework to a 6-DOF parafoil system to capture more complex dynamics, such as pitch and roll motions; (3) conducting rigorous field testing under diverse atmospheric conditions to validate simulation results and assess practical performance; and (4) performing a formal stability analysis, potentially using Lyapunov-based methods, to provide theoretical guarantees for the control system's robustness. These advancements aim to bridge the gap between simulation-based success and real-world deployment, further solidifying the NNMPC framework's utility in parafoil control.

Acknowledgement: We sincerely thank the College of Artificial Intelligence at Nankai University for providing essential computational resources that enabled the extensive simulations conducted in this study. We are also grateful for the insightful discussions with our colleagues, which significantly enriched the research direction and strengthened the outcomes of this work.

Funding Statement: This research was funded by the National Natural Science Foundation of China, grant number 62473210, awarded to J. Li. The funder's website can be accessed at https://www.nsfc.gov.cn/.

Author Contributions: Jie Li conceptualized and designed the study, developed the methodology, conducted the simulations, analyzed the data, and drafted the manuscript. Qinglin Sun contributed to validating the simulation results and provided supervision. Hao Sun supported resource provision and offered supervisory guidance. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Q. Sun, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Zhang L, Gao H, Chen Z, Sun Q, Zhang X. Multi-objective global optimal parafoil homing trajectory optimization via Gauss pseudospectral method. Nonlinear Dyn. 2013;72(1):1–8. doi:10.1007/s11071-012-0586-9.
2. Prakash O, Daftary A, Ananthkrishnan N. Trim and stability analysis of Parafoil/Payload system using bifurcation methods. In: 18th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar; 2005 May 23–26; Munich, Germany. doi:10.2514/6.2005-1666.
3. Zhang Z, Zhao Z, Fu Y. Dynamics analysis and simulation of six DOF parafoil system. Cluster Comput. 2019;22(5):12669–80. doi:10.1007/s10586-018-1720-3.
4. Li B, He Y, Han J, Xiao J. A new modeling scheme for powered parafoil unmanned aerial vehicle platforms: theory and experiments. Chin J Aeronaut. 2019;32(11):2466–79. doi:10.1016/j.cja.2019.04.001.
5. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. Automatica. 2002;38(1):3–20. doi:10.1016/S0005-1098(01)00174-1.
6. Maciejowski JM. Predictive control: with constraints. Hoboken, NJ, USA: Prentice-Hall; 1999.
7. Wu Y, Zuo Z, Wang Y, Han Q, Hu C. Distributed data-driven model predictive control for heterogeneous vehicular platoon with uncertain dynamics. IEEE Trans Veh Technol. 2023;72(8):9969–83. doi:10.1109/TVT.2023.3262705.
8. Pauca O, Caruntu CF, Lazar C. Predictive control for the lateral and longitudinal dynamics in automated vehicles. In: 23rd International Conference on System Theory, Control and Computing (ICSTCC); 2019 Oct 9–11; Sinaia, Romania: IEEE; 2019. p. 797–802. doi:10.1109/ICSTCC.2019.8885839.
9. Bao H, Kang Q, Shi X, Xiao L, An J. Robust learning-based model predictive control for intelligent vehicles with unknown dynamics and unbounded disturbances. IEEE Trans Intell Veh. 2024;9(2):3409–21. doi:10.1109/TIV.2023.3336964.
10. Wu H, Si Z, Li Z. Trajectory tracking control for four-wheel independent drive intelligent vehicle based on model predictive control. IEEE Access. 2020;8:73071–73081. doi:10.1109/ACCESS.2020.2987812.
11. Di Carlo J, Wensing PM, Katz B, Bledt G, Kim S. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2018 Oct 1–5; Madrid, Spain: IEEE. p. 1–9. doi:10.1109/IROS.2018.8594448.
12. Ding Y, Pandala A, Li C, Shin YH, Park HW. Representation-free model predictive control for dynamic motions in Quadrupeds. IEEE Trans Robot. 2021;37(4):1154–71. doi:10.1109/TRO.2020.3046415.
13. Chen D, Zhang J, Lv M, Chen Y. Research on model predictive control of quadruped robot based on improved single rigid body dynamics model. In: 2024 36th Chinese Control and Decision Conference (CCDC); 2024 May 25–27; Xi'an, China. p. 4294–9. doi:10.1109/CCDC62350.2024.10588014.
14. Iskandar M, van Ommeren C, Wu X, Albu-Schäffer A, Dietrich A. Model predictive control applied to different time-scale dynamics of flexible joint robots. IEEE Robot Autom Lett. 2023;8(2):672–9. doi:10.1109/LRA.2022.3229226.
15. Li X, Dong Z, Cao Y, Qin J, Zhang Z, Tse CK, et al. Model-predictive control with parameter identification for multi-dual-active-bridge converters achieving accurate power balancing. IEEE Trans Power Electron. 2023;38(9):10880–94. doi:10.1109/TPEL.2023.3290671.

16. Stano P, Montanaro U, Tavernini D, Tufo M, Fiengo G, Novella L, et al. Model predictive path tracking control for automated road vehicles: a review. Annu Rev Control. 2023;55(2):194–236. doi:10.1016/j.arcontrol.2022.11.001.

17. Son SH, Kim JW, Oh TH, Lee G, Lee JM. Improved offset-free model predictive control utilizing learned model-plant mismatch map. IFAC-PapersOnLine. 2022;55(7):792–7. doi:10.1016/j.ifacol.2022.07.541.

18. Baumeister T, Brunton SL, Kutz JN. Deep learning and model predictive control for self-tuning mode-locked lasers. J Opt Soc Am B. 2018;35(3):617–26. doi:10.1364/JOSAB.35.000617.

19. Clark NN. Neural network modeling of the emissions and performance of a heavy-duty diesel engine; 1999. [cited 2025 May 19]. Available from: https://api.semanticscholar.org/CorpusID:11513217.

20. Zhang C, Xu X, Zhang X, Zhou X, Lu Y, Zhang Y. A data-driven fault detection and diagnosis method via just-in-time learning for unmanned ground vehicles. Automatika. 2023;64(2):277–90. doi:10.1080/00051144.2022.2142924.

21. Di Natale L, Svetozarevic B, Heer P, Jones CN. Physically consistent neural networks for building thermal modeling: theory and analysis. Appl Energy. 2022;325(10):119806. doi:10.1016/j.apenergy.2022.119806.

22. Ishii K, Matsumoto H, Takeda M, Kawakami A, Yamada T. A high voltage intelligent power module (HVIPM) with a high performance gate driver. In: Proceedings of the 10th International Symposium on Power Semiconductor Devices and ICs. ISPSD’98 (IEEE Cat. No. 98CH36212); 1998 Jun 3–6; Kyoto, Japan. p. 289–92. doi:10.1109/ISPSD.1998.702690.

23. Lenz I, Knepper RA, Saxena A. DeepMPC: learning deep latent features for model predictive control. In: Robotics: Science and Systems; 2015 Jul 13–17; Roma, Italy.

24. Spielberg NA, Brown M, Kapania NR, Kegelman JC, Gerdes JC. Neural network vehicle models for high-performance automated driving. Sci Robot. 2019;4(28):eaaw1975. doi:10.1126/scirobotics.aaw1975.

25. Güven K, Şamiloğlu AT. System identification of an aerial delivery system with a ram-air parachute using a NARX network. Aerospace. 2022;9(8):443. doi:10.3390/aerospace9080443.

26. Zhao L, Tao J, Sun H, Sun Q. Dynamic modelling of parafoil system based on aerodynamic coefficients identification. Automatika. 2022;64(2):1–13. doi:10.1080/00051144.2022.2145697.

27. Gao H, Tao J, Dehmer M, Emmert-Streib F, Sun Q, Chen Z, et al. In-flight wind field identification and prediction of parafoil systems. Appl Sci. 2020;10(6):1958. doi:10.3390/app10061958.

28. Ward M, Costello M, Slegers N. Specialized system identification for parafoil and payload systems. J Guid Control Dyn. 2012;35(2):588–97. doi:10.2514/1.53364.

29. Spielberg NA, Brown M, Gerdes JC. Neural network model predictive motion control applied to automated driving with unknown friction. IEEE Trans Control Syst Technol. 2022;30(5):1934–45. doi:10.1109/TCST.2022.3167404.

30. Ni Y, Zhang W, Lv Y. Fast structural dynamic modeling and analysis for horizontally folding wing. Rev Int Métodos Numér Cálc Diseño Ing. 2021;37(3):35. doi:10.1007/s11831-021-09533-0.

31. Ni Y, Li P, Zhao H, Qiao S. Aeroelasticity analysis of folding wing with structural nonlinearity using modified DLM. Rev Int Métodos Numér Cálc Diseño Ing. 2024;40(3):36.

32. Shao J, Ling J, Zhang R, Cheng X, Zhang H. Research of an EPB shield pressure and depth prediction model based on deep neural network and its control device. Rev Int Métodos Numér Cálc Diseño Ing. 2024;40(1):1.

33. Liu C, Jiang R, Wei D, Yang C, Li Y, Wang F, et al. Deep learning approaches in flow visualization. Adv Aerodyn. 2022;4(1):17. doi:10.1186/s42774-022-00113-1.

34. Linot AJ, Zeng K, Graham MD. Turbulence control in plane Couette flow using low-dimensional neural ODE-based models and deep reinforcement learning. Int J Heat Fluid Flow. 2023;101:109139. doi:10.1016/j.ijheatfluidflow.2023.109139.

35. Balla K, Sevilla R, Hassan O, Morgan K. An application of neural networks to the prediction of aerodynamic coefficients of aerofoils and wings. Appl Math Model. 2021;96(10):456–79. doi:10.1016/j.apm.2021.03.019.
36. Yang H, Song L, Chen W. Research on parafoil stability using a rapid estimate model. Chin J Aeronaut. 2017;30(5):1670–80. doi:10.1016/j.cja.2017.06.003.
37. He W, Zheng Y, Wen J, Tao J, Tao J. Path following control of parafoil system based on SGLOS and improved ADRC tuned by MSMPA. IEEE Trans Circuits Syst II Express Briefs. 2024;71(8):3895–9. doi:10.1109/TCSII.2024.3377010.
38. Guo Y, Yan J, Xing X, Wu X, Li L. Predefined-time heading control for a 9-DOF parafoil recovery system subject to internal relative motions. Aerospace. 2023;10(4):348. doi:10.3390/aerospace10040348.
39. Liu Y, Wang X, Li J. Neural network-based model predictive control for a hybrid electric vehicle considering longitudinal and lateral dynamics. Aerosp Sci Technol. 2024;153:108484. doi:10.1016/j.ast.2024.108484.