drones

MDPI

*Article*

# Finite-Time Adaptive Reinforcement Learning Control for a Class of Morphing Unmanned Aircraft with Mismatched Disturbances and Coupled Uncertainties

**Wei Ren [1,2,*], Yingjie Wei [1], Cong Wang [1] and Zheng Wang [3]**

[1] School of Astronautics, Harbin Institute of Technology, Harbin 150001, China; yingjiewei@hit.edu.cn (Y.W.); alanwang@hit.edu.cn (C.W.)
[2] Jiangnan Electromechanical Design Institute, Guiyang 550009, China
[3] Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China
* Correspondence: 17b918103@stu.hit.edu.cn

## Highlights

### What are the main findings?

- A novel RL-based adaptive finite-time control scheme for morphing unmanned aircraft is synthesized. It can address mismatched disturbances, coupled uncertainties, and non-affine characteristics, enabling the aircraft's attitude to converge to the desired value within a finite time.
- The attitude dynamics of the morphing unmanned aircraft are described as a class of mismatched non-affine systems, which are more suitable for practical scenarios and simplify the analysis process compared to previous models.

### What is the implication of the main finding?

- Applying reinforcement learning to morphing unmanned aircraft enhances its ability to handle uncertainties, and finite-time reinforcement learning helps limit the control convergence time, thus improving the trajectory-tracking control performance.
- The proposed scheme for RL-based robust adaptive flight control offers a method that can be extended to various aircraft control research fields.

## Abstract

This paper proposes a finite-time adaptive reinforcement learning (RL) control law for a class of morphing unmanned aircraft with mismatched disturbances and coupled uncertainties. To handle the mismatched disturbances, an adaptive upper-bound estimator as well as the parameter adaptive laws have been proposed. Aiming at the coupled uncertainties, an RL-based online uncertainty estimator and a corresponding finite-time compensation control law have been developed. To deal with the non-affine characteristics, an auxiliary integral system has been introduced. By systematically integrating the aforementioned adaptive upper-bound estimators, finite-time control law, and the auxiliary signals, a novel RL-based adaptive finite-time control framework is constructed for morphing unmanned aircraft. Simulation results reveal the finite-time convergence and the advantages of the proposed method.

**Keywords:** reinforcement learning; finite-time adaptive control; morphing unmanned aircraft; mismatched disturbance; coupled uncertainties

check for updates

Academic Editor: Andrey V. Savkin

Received: 3 July 2025
Revised: 31 July 2025
Accepted: 6 August 2025
Published: 11 August 2025

**Citation:** Ren, W.; Wei, Y.; Wang, C.; Wang, Z. Finite-Time Adaptive Reinforcement Learning Control for a Class of Morphing Unmanned Aircraft with Mismatched Disturbances and Coupled Uncertainties. *Drones* **2025**, *9*, 562. https://doi.org/10.3390/drones9080562

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Morphing unmanned aircraft can optimize their aerodynamics and performance by changing their shape to adapt to diverse flight scenarios [1]. It can be classified by morphing scale, location, gas flow, and implementation method [2]. Yu et al. [3] and Noordin et al. [4] developed PID-based UAV adaptive control methods, achieving satisfactory tracking performance. As the application scenarios increase, there is an urgent need to improve its modeling and control law design [5]. Guidance and control technology is crucial as it ensures stable flight in complex environments, enables real-time adjustment of flight paths and morphing strategies, and enhances overall performance and adaptability [6,7]. He et al. [8] proposed an integrated guidance and control method using backstepping and fixed-time sliding mode control. This method includes a morphing term and solves the guidance and control problems of high-speed morphing aircraft. It stabilizes the system and makes errors converge quickly, thus improving control performance. Zhang et al. [9] presented an event-triggered fixed-time sliding mode control, which enhances the aircraft's robustness and adaptability under complex conditions. Abouheaf et al. [10] developed a machine-learning-based autonomous morphing control for flexible-wing morphing aircraft. This method shows better stability than conventional model-free adaptive control approaches. The literature [11,12] designs a feedback-linearization-based nonlinear command and stability system for variable aspect ratio morphing aircraft. This system improves flexibility and fuel efficiency. The literature [13] proposes an online actor–critic control for variable aspect-ratio and swept-morphing-wing aircraft, which solves the problem of control input frequency. Numerical simulations prove its effectiveness. The literature [14] proposes a preset-performance sliding-mode controller for high-speed morphing unmanned aircraft attitude control under strong uncertainty. Paired with a finite-time neural network disturbance observer, it enhances the system's performance and adaptability to disturbances and parameter changes. These methods combine advanced control theories to solve the high-speed guidance and control problems of morphing unmanned aircraft.

In recent years, advanced adaptive control approaches have been extensively researched and applied in the realm of complex and uncertain systems [15–17]. Among them, model reference adaptive control (MRAC) is a prominent technique that enables real-time parameter adjustment. It achieves this by comparing the performance of the system with a pre-established reference model. The literature [18–20] puts forward an enhanced nonlinear dynamic inversion control method based on MRAC. This innovation strengthens the aircraft's resilience against faults and external disturbances. It refines the accuracy of adaptive estimation and incorporates a low-pass filter to optimize the balance between dynamic performance and robustness. To design the controller better, it is crucial to elucidate the distinctions and relationships among parameter uncertainties, disturbance rejection, and fault-tolerant control. Parameter uncertainties can be caused by sensor biases, which introduce inaccuracies in the measurement of system states. Actuator failures, on the other hand, represent a severe type of disturbance. These failures can disrupt the normal operation of the aircraft and require effective fault-tolerant control strategies to ensure safe flight. To ensure robust performance, researchers focus on parameter uncertainties and disturbance rejection. For robustness enhancement, fuzzy adaptive control effectively strengthens system robustness under uncertain and fuzzy operating conditions. As detailed in the literature [21], a novel fixed-time adaptive generalized type-2 fuzzy logic control scheme was devised for hypersonic aircraft grappling with uncertainties. The efficacy of this control scheme is validated through simulations. Based on the observer, a robust flight controller for a disturbance-unmanned aerial vehicle system is developed [22]. In addition, in the literature [23], the control problem of nonuniform nonlinear systems with time-varying delays in both state and input has also been discussed. Adaptive sliding

mode control has also made significant strides in enhancing system robustness and responsiveness. In reference [24], a dedicated control strategy for hypersonic aircraft has been proposed. It constructs a dynamic uncertainty model and estimates uncertainty bounds online to coordinate the robustness and responsiveness of the controller. During flight, morphing unmanned aircraft experience complex shape alterations and fluctuations in aerodynamic characteristics. Traditional control methods like MRAC or sliding mode control often can't accurately capture these nonlinear dynamics. To address this, some scholars have studied the finite-time asynchronous control of singular fuzzy Markov jump systems [25]. Their research resulted in an adaptive event-triggered scheme, which reduces communication frequency and improves system efficiency. Additionally, Shi et al. [26] introduced a distributed adaptive event-triggered control strategy to solve the cooperative output regulation problem in heterogeneous linear multi-agent systems.

Nevertheless, abrupt changes in morphing unmanned aircraft, particularly shape-switching events, can surpass the predefined triggering conditions. This may result in overly frequent or missed trigger events, thereby introducing instability into the system. Adaptive neural network control emerges as a viable solution. It can manage nonlinear and high-dimensional systems, ensuring stability under unknown dynamics and external disturbances. Research [27,28] introduced a fault-tolerant control approach that combines an adaptive neural network with a nonlinear observer. This synergy improves the robustness and real-time decision-making capabilities of nonlinear systems. To optimize control performance, reinforcement learning has been integrated into the adaptive control framework, expanding its applicability. Reference [29] presented an adaptive model-free fault-tolerant control solution based on integral reinforcement learning for highly flexible aircraft with actuator failures. Reinforcement-learning-based methods, with their distinct advantages in handling nonlinearity, model-free scenarios, real-time learning, and multi-objective optimization, can surmount the control challenges of morphing unmanned aircraft in complex environments.

However, most existing design methods for proving the Lyapunov stability theory can only achieve asymptotic stability. That is, the system takes an infinite time to converge to the equilibrium point [30–33]. In many engineering applications, it is desired that the control objectives be achieved as soon as possible. Thus, finite-time control has emerged. Finite-time control is applicable to different types of systems. For example, references [34–36] proposed finite-time control methods for different types of nonlinear systems, ensuring the finite-time stability of the closed-loop systems. In the aerospace field, finite-time control is mainly used for spacecraft attitude tracking. For instance, reference [37] studied the vibration suppression and attitude tracking of flexible spacecraft under model uncertainties and external disturbances. It designs a multivariable finite-time control scheme for spacecraft attitude tracking based on a novel dynamic sliding dynamics and an adaptive disturbance observer (ADO). Reference [38] investigated the finite-time tracking control problem of a class of nonlinear systems and proposed a new finite-time command-filtered backstepping method. This method has the advantages of conventional command-filtered backstepping control and ensures finite-time convergence.

Research [39] designed a concurrent-learning adaptive finite-time control based on inertial parameter identification under external disturbances and applied it to aerospace attitude control. Reference [40] proposed a novel discrete-time fuzzy preselected performance control (PPC) method. By employing an indirect stabilization mechanism and a low-computation fuzzy approximation strategy, this approach achieves finite-time convergent control of unknown system dynamics without the need for complex model reconstruction. Furthermore, reference [41] developed a fixed-time pre-configured controller for electromechanical systems based on enhanced fuzzy neural approximation and backstep-

ping design. Extending this research to hypersonic vehicles with elevator-stuck faults, the authors propose a fuzzy fault-tolerant control scheme that ensures appointed-time convergence [42]. Reference [43] studied the spacecraft formation flying system affected by external disturbances and parameter uncertainties; it designed a new improved, fast integral terminal sliding-mode control law, and proposed an adaptive tracking control for the spacecraft formation flying system.

Only a few studies focus on applying finite-time control to morphing unmanned aircraft. In the early stage, Wang et al. [44] proposed a smooth-switching state-feedback controller design method for the altitude-keeping and attitude-stability problems of morphing unmanned aircraft during continuous morphing. They establish a chained smooth-switching system model and derive sufficient conditions for finite-time boundedness and robustness. Subsequently, Cheng et al. [45] studied the asynchronous finite-time H∞ control problem of morphing unmanned aircraft with controller uncertainties. Considering the inherent packet dropouts of the system and controller uncertainties, they propose a non-fragile finite-time H∞ controller design method, and its effectiveness is verified by numerical examples. In recent years, some scholars have combined finite-time control with disturbance observers to improve the robustness of morphing unmanned aircraft [46,47].

Based on the above analysis, the application of reinforcement learning to the trajectory-tracking control of morphing unmanned aircraft has not been reported yet. In particular, the research on finite-time reinforcement learning in this field is lacking. Considering the strong coupling and uncertainties of morphing unmanned aircraft, the application of reinforcement learning can enhance its ability to deal with uncertainties. Meanwhile, finite-time reinforcement learning can help limit and shorten the system's stabilization and control convergence time. Therefore, it is of great significance to improve the trajectory-tracking control performance of morphing unmanned aircraft, and this paper will conduct in-depth research on this.

Based on the above analysis, the main contributions of this paper are as follows:

1. The attitude dynamics of the morphing unmanned aircraft are described as a class of mismatched non-affine systems, including matched and mismatched disturbances, non-affine input, and internal uncertainties. Compared to previous models [14,48,49], the proposed model is more applicable to practical scenarios and simplifies the analysis process.
2. Compared with the literature [13], our work focuses on finite-time control, mismatched disturbances, and coupled uncertainties, while the literature [13] addresses non-affine control systems and control input frequency constraints. Different from the literature [27] on highly flexible aircraft, our method targets morphing unmanned aircraft, devises adaptive finite-time controllers, and ensures finite-time attitude convergence with better performance.
3. This paper proposes a design framework for RL-based adaptive anti-disturbance flight control, offering a paradigm that can be extended to various aircraft.

The remainder of this paper is structured as follows. In Section 2, the problem is formulated, and preliminary knowledge is introduced. Section 3 presents the main findings. Section 4 provides simulation studies. Finally, Section 5 concludes the paper.

## 2. Problem Formulation and Preliminaries

### *2.1. Dynamic Model Description*

The attitude dynamics model of a morphing unmanned aircraft can be formulated as follows [8,14]:

$$\begin{aligned}\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}_v\end{bmatrix} &= \begin{bmatrix}-\cos\alpha\tan\beta & \sin\alpha\tan\beta & 1\\ \sin\alpha & \cos\alpha & 0\\ \cos\alpha\sec\beta & -\sin\alpha\sec\beta & 0\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix}\\ -\begin{bmatrix}\sec\beta\cos\gamma_v\\ \sin\gamma_v\\ -\tan\beta\cos\gamma_v\end{bmatrix}\dot{\theta} &- \begin{bmatrix}-\sec\beta\cos\theta\sin\gamma_v\\ \cos\theta\cos\gamma_v\\ -\sin\theta+\tan\beta\cos\theta\sin\gamma_v\end{bmatrix}\dot{\sigma},\end{aligned} \tag{1}$$

where $\alpha$ is the angle of attack, $\beta$ is the sideslip angle, $\gamma_v$ is the tilt angle, $\theta$ is the velocity inclination angle, and $\sigma$ is the track yaw angle, and $\omega_x, \omega_y, \omega_z$ are the angular velocities of the roll, yaw, and pitch channels, respectively.

The longitudinal dynamics model of the aircraft is as follows:

$$\dot{\boldsymbol{\omega}} = \left(\mathbf{I}_b + \sum_{j=L,r}\mathbf{I}_{bj}\right)^{-1}\left(\mathbf{M}_c - \mathbf{M}_f - \sum_{j=L,r} m_j\left(\mathbf{r}_{Om\to Oj} + \mathbf{r}_{Oj\to c_j}\right)\times(\mathbf{v}+\boldsymbol{\omega}\times\mathbf{v})\right), \tag{2}$$

where $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$, $\mathbf{I}_b$ denotes the moment of inertia of the aircraft body, $\mathbf{I}_{bj}$ denotes the moment of inertia of the wing, $\mathbf{M}_c$ represents the aerodynamic torque associated with control ability, $\mathbf{M}_f$ is the additional moment caused by deformation, $\mathbf{r}_{Om\to Oj}$ is the rotation vector of the wing relative to the fuselage in this system, $\mathbf{r}_{Oj\to c_j}$ is the rotation vector of the wing relative to the fuselage under the wing mounting system, $\mathbf{v}$ represents the matrix of velocity magnitudes, and $m_j$ is the mass of the aircraft wing, respectively. The detailed descriptions of some notations are as follows.

The $\mathbf{I}_{bj}$ is expressed as:

$$\begin{aligned}\mathbf{I}_{bj} = &-m_j\mathbf{r}^{\times}_{Om\to Oj}\times\mathbf{r}^{\times}_{Om\to Oj} - m_j\mathbf{R}_{bj}\mathbf{r}^{\times}_{Oj\to c_j}\times\mathbf{R}^T_{bj}\mathbf{r}^{\times}_{Om\to Oj}\\ &-m_j\mathbf{r}^{\times}_{Om\to Oj}\mathbf{R}_{bj}\mathbf{r}^{\times}_{Oj\to c_j}\times\mathbf{R}^T_{bj}\mathbf{r}^{\times}_{Om\to Oj} + \mathbf{R}_{bj}I_j\mathbf{R}^T_{bj},\end{aligned} \tag{3}$$

where $\mathbf{r}^{\times}_{Oj\to c_j}$ and $\mathbf{r}^{\times}_{Om\to Oj}$ represent the cross-product vector matrices of $\mathbf{r}_{Oj\to c_j}$ and $\mathbf{r}_{Om\to Oj}$, respectively, $\mathbf{R}_{bj}$ is a rotation matrix from the wing installation frame to the body coordinate system, and is expressed as follows:

$$\begin{cases}\mathbf{R}_{bl} = \left[\ \mathbf{R}_y(90^\circ+\chi_l)\ \right]^T\\ \mathbf{R}_{br} = \left[\ \mathbf{R}_y(-90^\circ-\chi_r)\ \right]^T,\end{cases} \tag{4}$$

where $\mathbf{R}_y$ is the basic rotation matrix of the $y$ axis, and $\chi$ is the caster angle.

The $\mathbf{M}_c$ can be given as follows:

$$\mathbf{M}_c = \begin{bmatrix}M_{cx}\\ M_{cy}\\ M_{cz}\end{bmatrix} = \frac{1}{2}\rho v^2 S_0 L_0\begin{bmatrix}m_{cx}\\ m_{cy}\\ m_{cz}\end{bmatrix}, \tag{5}$$

where $\rho$ represents the atmospheric density, $v$ is the velocity magnitude, $S_0$ represents the reference area, $L_0$ is the reference length, $m_{cx}, m_{cy}, m_{cz}$ represents the rotation moment coefficient, yaw moment coefficient, and pitch moment coefficient, respectively.

The determination of $m_{ci}$ can be presented by

$$m_{ci} = m_{ci}(\alpha, \beta, \chi, \delta_x, \delta_y, \delta_z), \quad i = x, y, z, \tag{6}$$

where $\delta_x$, $\delta_y$ and $\delta_z$ represent the roll angle, yaw angle, and pitch angle, respectively.

The $\mathbf{M}_f$ can be given as follows:

$$\begin{aligned}\mathbf{M}_{\mathrm{f}} &= \boldsymbol{\omega}_b \times (\mathbf{I}_b \boldsymbol{\omega}_b) + \sum_{j=L,r} \mathbf{I}_{bj} \dot{\boldsymbol{\omega}}_j + \sum_{j=L,r} m_j \mathbf{r}_{Om \to Oj} \times \left[\boldsymbol{\omega}_b \times \left(\boldsymbol{\omega}_b \times \mathbf{r}_{Om \to Oj}\right)\right] \\ &+ \sum_{j=L,r} m_j \mathbf{r}_{Oj \to c_j} \times \left[\boldsymbol{\omega}_b \times \left(\boldsymbol{\omega}_b \times \mathbf{r}_{O_m \to O_j}\right)\right] \\ &+ \sum_{j=L,r} m_j \mathbf{r}_{O_m \to O_j} \times \left\{\left(\boldsymbol{\omega}_b + \boldsymbol{\omega}_j\right) \times \left[\left(\boldsymbol{\omega}_b + \boldsymbol{\omega}_j\right) \times \mathbf{r}_{Oj \to c_j}\right]\right\},\end{aligned} \tag{7}$$

where $\mathbf{I}_{bj} = \mathbf{R}_{bj} I_b \mathbf{R}_{bj}^T$, $\boldsymbol{\omega}_b$ represents the angular velocity around the center of mass $O_m$ of the aircraft in the body coordinate system, $\boldsymbol{\omega}_j$ and $\dot{\boldsymbol{\omega}}_j$ are the angular velocity and angular acceleration, respectively.

By defining the state variables $\boldsymbol{x}_1 = \begin{bmatrix} x_{11} & x_{12} & x_{13} \end{bmatrix} = \begin{bmatrix} \alpha & \beta & \gamma_v \end{bmatrix}^T$ and $\boldsymbol{x}_2 = \begin{bmatrix} x_{21} & x_{22} & x_{23} \end{bmatrix}^T = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$, control input $\boldsymbol{u} = \begin{bmatrix} \delta_x & \delta_y & \delta_z \end{bmatrix}^T$, then (1) and (2) can be simplified in the following vector form with uncertain mismatched non-affine functions:

$$\begin{aligned}\dot{\boldsymbol{x}}_1(t) &= \boldsymbol{f}_1(\boldsymbol{x}_1(t)) + \Delta \boldsymbol{f}_1(\boldsymbol{x}_1(t)) + \boldsymbol{g}_1(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)) + \boldsymbol{d}_1(t) \\ \dot{\boldsymbol{x}}_2(t) &= \boldsymbol{f}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)) + \Delta \boldsymbol{f}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)) + \boldsymbol{g}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \boldsymbol{u}(t)) + \boldsymbol{d}_2(t),\end{aligned} \tag{8}$$

where $\Delta \boldsymbol{f}_1(\boldsymbol{x}_1(t))$ and $\Delta \boldsymbol{f}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$ are the unknown structural uncertainties caused by the deformation of the aircraft, $\boldsymbol{d}_1(t)$ and $\boldsymbol{d}_2(t)$ represent the various unknown disturbances encountered during flight, $\boldsymbol{g}_1(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$ and $\boldsymbol{g}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \boldsymbol{u}(t))$ denotes the non-affine functions, some notations are detailed as:

$$\begin{aligned}\boldsymbol{f}_1(\boldsymbol{x}_1(t)) &= -\begin{bmatrix} \sec\beta\cos\gamma_v & \sin\gamma_v & -\tan\beta\cos\gamma_v \end{bmatrix}^T \dot{\theta} \\ &-\begin{bmatrix} -\sec\beta\cos\theta\sin\gamma_v & \cos\theta\cos\gamma_v & -\sin\theta + \tan\beta\cos\theta\sin\gamma_v \end{bmatrix}^T \dot{\sigma},\end{aligned} \tag{9}$$

$$\begin{aligned}\boldsymbol{f}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)) &= -\left(\boldsymbol{I}_b + \sum_{j=L,r} \boldsymbol{I}_{bj}\right)^{-1} \boldsymbol{M}_{\mathrm{f}} \\ &-\left(\boldsymbol{I}_b + \sum_{j=L,r} \boldsymbol{I}_{bj}\right)^{-1} \left(\sum_{j=L,r} m_j \left(\boldsymbol{r}_{Om \to Oj} + \boldsymbol{r}_{Oj \to c_j}\right) \times (\boldsymbol{v} + \boldsymbol{\omega} \times \boldsymbol{v})\right).\end{aligned} \tag{10}$$

$$\begin{aligned}\boldsymbol{g}_1(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)) &= \begin{bmatrix} -\cos\alpha\tan\beta & \sin\alpha\tan\beta & 1 \\ \sin\alpha & \cos\alpha & 0 \\ \cos\alpha\sec\beta & -\sin\alpha\sec\beta & 0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \\ \boldsymbol{g}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \boldsymbol{u}(t)) &= \left(\boldsymbol{I}_b + \sum_{j=L,r} \boldsymbol{I}_{bj}\right)^{-1} \begin{bmatrix} m_{cx}(\alpha, \beta, \chi, \delta_x, \delta_y, \delta_z) \\ m_{cy}(\alpha, \beta, \chi, \delta_x, \delta_y, \delta_z) \\ m_{cz}(\alpha, \beta, \chi, \delta_x, \delta_y, \delta_z) \end{bmatrix}.\end{aligned} \tag{11}$$

**Remark 1.** *In the research of morphing unmanned aircraft control system modeling, to conform to real-world engineering practices and simplify the analysis, specific symbols are defined.* $\Delta \boldsymbol{f}_1(\boldsymbol{x}_1(t))$ *and* $\Delta \boldsymbol{f}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$ *stand for unknown structural uncertainties due to aircraft deformation,* $\boldsymbol{d}_1(t)$ *and* $\boldsymbol{d}_2(t)$ *denote various unknown flight disturbances. These flight disturbances encompass sensor biases that can lead to parameter uncertainties, as well as actuator failures, which are a particularly severe type of disturbance. Additionally,* $\boldsymbol{g}_1(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$ and $\boldsymbol{g}_2(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \boldsymbol{u}(t))$

*represents non-affine functions. Furthermore, analysis of (9)–(11) shows that system (8) has significant non-affine features, posing great challenges to morphing unmanned aircraft control design.*

Thereafter, when there is no risk of confusion, vectors may not be bolded, and function arguments may be omitted.

Based on the above analysis and simplification, the control-oriented morphing unmanned aircraft attitude model is obtained as an uncertain mismatched non-affine system as follows:

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t)) + \Delta f_1(x_1(t)) + g_1(x_1(t), x_2(t)) + d_1(t) \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t)) + \Delta f_2(x_1(t), x_2(t)) + g_2(x_1(t), x_2(t), u(t)) + d_2(t) \\ y(t) &= x_1(t), \end{aligned} \tag{12}$$

where $y(t)$ represents the system output vector.

To design an RL-based adaptive fault-tolerant controller for a morphing unmanned aircraft facing external disturbances, unknown dynamics, and non-affine inputs, this paper sets two control goals: (a) All the signals in the closed-loop system are semi-global practical finite-time stable. (b) The racking errors converge to a small neighborhood of origin in a finite time.

The following assumptions and lemmas are necessary to design the controller.

**Assumption 1.** *To ensure the validity of the input, it is assumed that $\partial g_1/\partial x_1, \partial g_1/\partial x_2, \partial g_2/\partial x_1, \partial g_2/\partial x_2, \partial g_2/\partial u$ are all invertible matrices, and it has [50,51]:*

$$\lambda\left[\frac{\partial g_i}{\partial x_1}\right] \le \underline{\pi},\ \lambda\left[\frac{\partial g_i}{\partial x_2}\right] \le \underline{\pi},\ \lambda\left[\frac{\partial g_2}{\partial u}\right] \le \underline{\pi}, i = 1, 2, \tag{13}$$

*where $\lambda[\partial g_i/\partial x_1]$, $\lambda[\partial g_i/\partial x_2]$ and $\lambda[\partial g_2/\partial u]$ represents the eigenvalue of the matrix $\partial g_i/\partial x_1$, $\partial g_i/\partial x_2$ and $\partial g_2/\partial u$, respectively.*

**Assumption 2.** *The desired trajectory $y_d$ and its first derivative are continuous and bounded* [52].

**Definition 1.** *If for all $\epsilon(t_0) = \epsilon_0$, there exist $\updownarrow > 0$ and a settling time $T(\updownarrow, \epsilon_0) < \infty$ such that $\|\epsilon(t)\| < \updownarrow$ holds for all $t \ge t_0 + T$, then the equilibrium $\epsilon = 0$ of the nonlinear system $\dot{\epsilon} = f(\epsilon)$ is semi-global practical finite-time stable* [53].

**Lemma 1.** *For any $\lambda > 0$ and $x \in R$, there exists* [49,53]:

$$0 \le |x| - x\tanh\left(\frac{x}{\bar{\lambda}}\right) \le \kappa\lambda, \tag{14}$$

*where $\kappa = 0.2785$.*

**Lemma 2.** *Given any constants $\Im > 0$, $0 < l < 1$, and $\wp > 0$, consider system $\dot{\epsilon} = f(\epsilon)$* [54]. *If there exists a smooth positive-definite function $V(\eta)$ such that:*

$$\dot{V}(\epsilon) \le -\Im V^l(\epsilon) + \wp, t \ge 0, \tag{15}$$

*holds, then system $\dot{\epsilon} = f(\epsilon)$ is semi-global practical finite-time stable.*

**Lemma 3.** *For any real variables $p, q$ and any positive constants $a_1, a_2, \hbar$, the following inequality holds* [55,56]:

$$|p|^{a_1}|q|^{a_2} \le \frac{a_1}{a_1 + a_2}\hbar|p|^{a_1+a_2} + \frac{a_2}{a_1 + a_2}\hbar^{-\frac{a_1}{a_2}}|q|^{a_1+a_2}. \tag{16}$$

**Lemma 4.** *For $Y_i \in R$ and $0 < \gamma \leq 1$, the following inequality holds [57]:*

$$\left(\sum_{i=1}^{n}|Y_i|\right)^{\gamma} \leq \sum_{i=1}^{n}|Y_i|^{\gamma} \leq n^{1-\gamma}\left(\sum_{i=1}^{n}|Y_i|\right)^{\gamma}. \tag{17}$$

### 2.2. Designs of Actor–Critic Neural Networks

Research shows that radial basis function neural networks (RBFNNs) can approximate any smooth continuous function $f(x)$ with any precision [58]. As a current research hotspot, the actor–critic framework enables the critic network to receive information from the environment and evaluate control performance through a cost function. Based on this evaluation, the actor network generates a control strategy for the actuator. Owing to the incorporation of reinforcement signals, the resulting control law exhibits faster convergence and reduced steady-state error. Based on this, an RL framework is developed in this study, using RBFNNs to approximate unknown nonlinear functions and penalty functions, namely:

$$f(x) = W_a^{*T}\Phi_a(x) + \varepsilon_a(x), |\varepsilon_a(x)| \leq \bar{\varepsilon}_a, \tag{18}$$

where $x = [x_1, x_2, \ldots x_n]^T$ is the input vector of the RBFNN, $\varepsilon(x)$ is the approximation error, $W_a^*$ is the optimized vector of the weight vector $W_a = \left[\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_p\right]^T$, and $\Phi_a(x) = \left[\Phi_1(x), \ldots, \Phi_p(x)\right]^T$ is the basis function vector, $p$ is the number of hidden nodes, $\Phi_j(x) = e^{-(x-\mu_j)^2/(2\sigma_j{}^2)}, j = 1, \ldots, p$ is the basis function, $\mu_j$ is the center of the basis function, $\sigma_j$ is the width of the Gaussian basis function, $\varepsilon_a(x)$ is the approximation error, and $\bar{\varepsilon}_a$ is a positive integer.

When designing the actor–critic neural networks, several important design principles of the weight adaptive law in reinforcement learning are considered to ensure the effectiveness and stability of the system.

To approximate the unknown nonlinearity $\Delta f_i(\overline{x}_i)$ in (12), an actor neural network $W_{ai}^T\Phi_{ai}(\overline{x}_i)$ is introduced, which is expressed as follows:

$$\Delta f_i(\overline{x}_i) = W_{ai}^T\Phi_{ai}(\overline{x}_i) + \varepsilon_{ai}, \tag{19}$$

where $\overline{x}_i = [x_1, \ldots x_i]^T \in R^i, i = 1, 2$, $W_{ai}$ is the ideal actor neural network weight.

Define the weight error of the actor neural network as $\widetilde{W}_{ai} = \hat{W}_{ai} - W_{ai}$, then its approximation error is expressed as:

$$H_{ai} = \widetilde{W}_{ai}^T\Phi_{ai}(\overline{x}_i). \tag{20}$$

To improve the tracking performance, a new error function of the actor neural network is constructed based on the approximation error and the penalty function:

$$\begin{aligned} e_{ai} &= H_{ai} + \Gamma_i\hat{J}_i, \\ E_{ai} &= \tfrac{1}{2}e_{ai}^T e_{ai}, \end{aligned} \tag{21}$$

where $\Gamma_i$ is the gain to be chosen, and the goal of updating the network weights to minimize $E_{ai}$.

Based on the information gathered from the environment, an integral penalty function is constructed to generate RL signals:

$$J_i(t) = \int_{\tau}^{\infty} q_i(t)dt, \tag{22}$$

where $q_i(t) = z_i^T Q_i z_i$.

The specific meaning of the variables will be given later. The penalty function can be approximated by a critic neural network:

$$\begin{aligned} J_i &= W_{ci}^T \Phi_{ci}(\overline{x}_i) + \varepsilon_{ci}, \\ \hat{J}_i &= \hat{W}_{ci}^T \Phi_{ci}(\overline{x}_i), \end{aligned} \tag{23}$$

where $W_{ci}$ is the ideal critic neural network weight.

The update rule for the weights of the actor–critic network is designed as follows:

$$\dot{\hat{W}}_{ai} = -\tau_i \eta_i \hat{W}_{ai} - \eta_i \Phi_{ai}(\overline{x}_i) \left[ \Phi_{ai}^T \hat{W}_{ai} + \hat{J}_i \Gamma_i^T \right], \tag{24}$$

where $\eta_i > 0, \tau_i > 0$ are the parameters to be chosen.

Construct the residual mean square error function of the critic network:

$$\begin{aligned} e_{ci} &= q_i(t) + \dot{\hat{J}}_i = q_i(t) + \hat{W}_{ci}^T \dot{\Phi}_{ci}(\overline{x}_i), \\ E_{ci} &= \tfrac{1}{2} e_{ci}^T e_{ci}. \end{aligned} \tag{25}$$

The update rule for the weights of the critic network is derived using the gradient descent method as follows:

$$\dot{\hat{W}}_{ci} = -\omega_i \left( \hat{W}_{ci}^T \dot{\Phi}_{ci} + q_i(t) \right) \dot{\Phi}_{ci} - \omega_i \varpi_i \hat{W}_{ci}, \tag{26}$$

where $\omega_i$ and $\varpi_i$ are the positive design constants.

By applying these design principles, we can design an effective weight adaptive law for the actor–critic neural networks, which helps to improve the performance of the control system.

**Remark 2.** *The $\hat{J}$ term approximates the unknown value function, enabling the actor to consider future rewards. This has helped the actor learn an optimal long-term policy, as focusing only on immediate rewards would lead to sub-optimal behavior. The second correction term enhances the stability and convergence of the actor–critic algorithm. It can adjust the actor's update step, preventing aggressive updates that could cause divergence or oscillation. Moreover, it accounts for environmental uncertainties, improving the system's robust ability to handle unexpected situations.*

**Property 1.** *To satisfy the persistence excitation condition, the basis functions $\Phi_{ci}$ and $\Phi_{ai}$, and the derivatives $\dot{\Phi}_{ci}$ and $\dot{\Phi}_{ai}$ are assumed to satisfy $\left\| \dot{\Phi}_{ai} \right\| \leq \Phi_{aim}$, $\left\| \dot{\Phi}_{ci} \right\| \leq \Phi_{cim}$, $\|\Phi_{ai}\| \leq \Phi_{aiM}$, $\|\Phi_{ci}\| \leq \Phi_{ciM}, i = 1, 2$. Meanwhile, the estimation errors of actor–critic learning and their derivative are bounded, that is $|\varepsilon_{ci}| \leq \varepsilon_{ciM}$ and $\left|\dot{\varepsilon}_{ci}\right| \leq \varepsilon_{cim}$ [51].*

**Remark 3.** *RBFNNs are selected over other function approximations for several well-founded reasons. Firstly, owing to their universal approximation property, RBFNNs are capable of accurately approximating uncertainties [52]. Secondly, for high-dimensional data, RBFNNs offer computational efficiency as their training process is relatively straightforward.*

**Remark 4.** *When determining the centers and widths, a trial-and-error technique is used to process the input data. The centers are set in high-density regions, and prior knowledge of the system's critical operating points is incorporated. The widths are set according to the distances between the centers and fine-tuned on the validation set.*

**Remark 5.** *The penalty function (22) and update laws (24) and (26) are designed with the aim of ensuring the stability and convergence of the RL framework. The penalty function serves to penalize*

*undesired system behaviors, and the update laws are derived from Lyapunov techniques. The detailed derivations of these aspects will be presented in the revised manuscript.*

**Remark 6.** *The actor–critic neural network and RBFNN architectures, learning mechanisms, and generalization abilities are designed to suit different application scenarios. The actor–critic is for dynamic scenarios, while RBFNN excels at local function approximation.*

## 3. Main Results

This section details the design process of the RL-based adaptive fault-tolerant control method. First, by introducing an auxiliary integral term, the system is reformulated as an augmented affine system with the overall non-affine function treated as the control input. Next, a virtual control law is designed, which integrates the actor–critic network and the disturbance boundary estimator. The final control signal is obtained through recursive design, and the actual control input is generated through integration. The structure of the controller is shown in Figure 1.

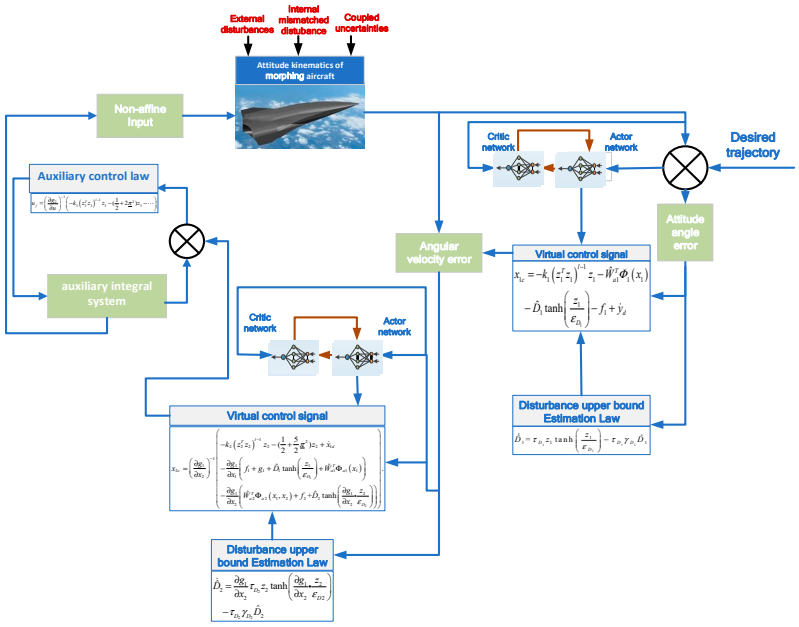



**Figure 1.** The controller structure.

### *3.1. The Design of the Augmented System*

Through the auxiliary integration method, the augmented system is established as follows:

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t)) + \Delta f_1(x_1(t)) + g_1(x_1(t), x_2(t)) + d_1(t) \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t)) + \Delta f_2(x_1(t), x_2(t)) + g_2(x_1(t), x_2(t), u(t)) + d_2(t) \\ \dot{u} &= u_f \\ y(t) &= x_1(t), \end{aligned} \tag{27}$$

where $d_1(t)$ denotes mismatched disturbance, $d_2(t)$ denotes matched disturbance, which are compensated by the Tanh function boundary estimator. Additionally, unknown nonlinear functions $\Delta f_1(x_1(t)), \Delta f_2(x_1(t), x_2(t))$ are approximated using an actor–critic framework.

Different from the affine system, non-affine systems treat the non-affine function holistically as the subsystem input, with the tracking error formally defined as follows:

$$\begin{aligned} z_1(t) &= x_1(t) - y_d \\ z_2(t) &= g_1(x_1(t), x_2(t)) - x_{1d}(t) \\ z_3(t) &= g_2(x_1(t), x_2(t), u(t)) - x_{2d}(t), \end{aligned} \tag{28}$$

where $y_d$ is the desired output value of the system, $x_{1d}$ and $x_{2d}$, the constructed virtual control variables, will be presented below.

Let $x_{ic}$ denote the virtual control law to be designed for the i-th subsystem. To mitigate differential explosion, a first-order low-pass filter [58] is introduced as:

$$\delta_i \dot{x}_{id} + x_{id} = x_{ic}, x_{id}(0) = x_{ic}(0), i = 1, 2, \tag{29}$$

where $0 < \delta_i < 1$ denotes the filter time constant to be designed, through which both $x_{id}$ and its derivative can be obtained. The boundary layer error is defined as:

$$y_i = x_{id} - x_{ic}, \tag{30}$$

where $x_{ic}$ denotes the filtered value of the virtual controller $x_{id}$.

Then we can obtain:

$$\dot{y}_i = \dot{x}_{id} - \dot{x}_{ic} = -\frac{y_i}{\delta_i} - \dot{x}_{ic}. \tag{31}$$

### *3.2. Controller Design*

**Step 1.** From the first equality in Equation (28), it yields:

$$\begin{aligned} \dot{z}_1(t) &= \dot{x}_1(t) - \dot{y}_d \\ &= f_1(x_1(t)) + \Delta f_1(x_1(t)) + g_1(x_1(t), x_2(t)) + d_1(t) - \dot{y}_d \\ &= f_1 + z_2 + y_1 + x_{1c} + d_1 + \Delta f_1 - \dot{y}_d. \end{aligned} \tag{32}$$

Let $\Delta \hat{f}_1$ denote the estimated value of the inner-loop nonlinearity $\Delta f_1$. According to Section 2.2, it can be approximated using an actor network, expressed as:

$$\Delta f_1 = W_{a1}^T \Phi_{a1}(x_1) + \varepsilon_{a1}, \tag{33}$$

where $\varepsilon_{a1}$ is the error of the actor network.

Define $D_1 = \sup_{t \geq 0} \| d_1(t) + \varepsilon_{a_1} \|$, thus, the virtual controller can be designed as:

$$x_{1c} = -k_1 \left( z_1^T z_1 \right)^{l-1} z_1 - \hat{W}_{a1}^T \Phi_1(x_1) - \hat{D}_1 \tanh\left( \frac{z_1}{\varepsilon_{D_1}} \right) - f_1 + \dot{y}_d. \tag{34}$$

Let $\hat{D}_1$ denote the estimated value of the mismatched disturbance. From Equations (32) and (34), we obtain:

$$z_1^T \dot{z}_1 = -k_1 \left( z_1^T z_1 \right)^l + z_1^T (z_2 + y_1) + z_1^T \left( d_1 + \varepsilon_{a1} - \hat{D}_1 \tanh\left( \tfrac{z_1}{\varepsilon_{D_1}} \right) \right) - z_1^T \widetilde{W}_{a1}^T \Phi_{a1}(x_1), \tag{35}$$

where $\widetilde{W}_{a1} = \hat{W}_{a1} - W_{a1}$.

The integral penalty function of the control system is defined as: $J_1(t) = \int_\tau^\infty q_1(t) dt$, where $q_1(t) = {z_1}^T Q_1 z_1$. Then, the actor–critic network weight updating law is designed as:

$$\dot{\hat{W}}_{a1} = -\tau_1 \eta_1 \hat{W}_{a1} - \eta_1 \Phi_{a1}(x_1) [{\Phi_{a1}}^T \hat{W}_{a1} + \hat{J}_1 \Gamma_1^T] \,. \tag{36}$$

The critic network weight updating law is derived via the gradient descent method as:

$$\dot{\hat{W}}_{c1} = -\varpi_1(\dot{\hat{W}}_{c1}^{T}\dot{\Phi}_{c1} + q_1(t))\dot{\Phi}_{c1} - \omega_1\varpi_1\hat{W}_{c1}. \tag{37}$$

The updating law of $\hat{D}_1$ is given as:

$$\dot{\hat{D}}_1 = \tau_{D_1} z_1 \tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right) - \tau_{D_1}\gamma_{D_1}\hat{D}_1. \tag{38}$$

**Step 2.** Define $D_2 = \sup_{t\geq 0}\|d_2(t) + \varepsilon_{a2}\|$, where $\hat{D}_2$ is the estimated value. According to Equation (28), we have:

$$\begin{aligned} \dot{z}_2 &= \dot{g}_1 - \dot{x}_{1d} \\ &= \tfrac{\partial g_1}{\partial x_1}\dot{x}_1 + \tfrac{\partial g_1}{\partial x_2}\dot{x}_2 - \dot{x}_{1d} \\ &= \tfrac{\partial g_1}{\partial x_1}(f_1 + g_1 + d_1 + \Delta f_1) + \tfrac{\partial g_1}{\partial x_2}(f_2 + \Delta f_2 + z_3 + y_2 + x_{2c} + d_2) - \dot{x}_{1d}. \end{aligned} \tag{39}$$

Introduce the actor network to compensate $\Delta f_2(t, x_1, x_2)$:

$$\begin{aligned} \Delta f_2(t, x_1, x_2) &= W_{a2}^{T}\Phi_{a2}(x_1, x_2) + \varepsilon_{a2}, \\ \Delta\hat{f}_2(t, x_1, x_2) &= \hat{W}_{a2}^{T}\Phi_{a2}(x_1, x_2). \end{aligned} \tag{40}$$

The designed virtual control law is:

$$x_{2c} = \left(\tfrac{\partial g_1}{\partial x_2}\right)^{-1}\begin{pmatrix} -k_2\left(z_2^{T}z_2\right)^{l-1}z_2 - (\frac{1}{2} + \frac{5}{2}\underline{\pi}^2)z_2 - \frac{\partial g_1}{\partial x_1}\left(f_1 + g_1 + \hat{D}_1\tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right) + \hat{W}_{a1}^{T}\Phi_{a1}(x_1)\right) \\ -\frac{\partial g_1}{\partial x_2}\left(\hat{W}_{a2}^{T}\Phi_{a2}(x_1, x_2) + f_2 + \hat{D}_2\tanh\left(\frac{\partial g_1}{\partial x_2}\cdot\frac{z_2}{\varepsilon_{D_2}}\right)\right) + \dot{x}_{1d} \end{pmatrix}. \tag{41}$$

Substituting into Equation (39), it yields:

$$\begin{aligned} z_2^{T}\dot{z}_2 = -k_2\left(z_2^{T}z_2\right)^{l} + \tfrac{\partial g_1}{\partial x_1}z_2\left(d_1 + \varepsilon_{a_1} - \hat{D}_1\tanh\left(\tfrac{z_1}{\varepsilon_{D_1}}\right)\right) - \tfrac{\partial g_1}{\partial x_1}z_2\widetilde{W}_{a1}^{T}\Phi_{a1}(x_1) + \tfrac{\partial g_1}{\partial x_2}z_2(z_3 + y_2) \\ -\tfrac{\partial g_1}{\partial x_2}z_2\widetilde{W}_{a2}^{T}\Phi_{a2}(x_1, x_2) + \tfrac{\partial g_1}{\partial x_2}z_2\left(d_2 + \varepsilon_{a_2} - \hat{D}_2\tanh\left(\tfrac{\partial g_1}{\partial x_2}\cdot\tfrac{z_2}{\varepsilon_{D_2}}\right)\right) \end{aligned} \tag{42}$$

Similarly, the actor–critic framework is designed with the penalty function defined as: $J_2(t) = \int_{\tau}^{\infty} q_2(t)dt$ and $q_2(t) = {z_2}^{T}Q_2z_2$.

Construct the critic network module effect error function and design the weight update law as:

$$\begin{aligned} \dot{\hat{W}}_{a2} &= -\tfrac{\partial g_1}{\partial x_2}\eta_2\Phi_{a2}(x_2)[\tfrac{\partial g_1}{\partial x_2}{\Phi_{a2}}^{T}\hat{W}_{a2} + \hat{f}_2\Gamma_2^{T}] - \eta_2\tau_2\hat{W}_{a2}, \\ \dot{\hat{W}}_{c2} &= -\varpi_2(\dot{\hat{W}}_{c2}^{T}\dot{\Phi}_{c2} + q_2(t))\dot{\Phi}_{c2} - \omega_2\varpi_2\hat{W}_{c2}. \end{aligned} \tag{43}$$

The updating law of $\hat{D}_2$ is given as:

$$\dot{\hat{D}}_2 = \frac{\partial g_1}{\partial x_2}\tau_{D_2}z_2\tanh\left(\frac{\partial g_1}{\partial x_2}\cdot\frac{z_2}{\varepsilon_{D2}}\right) - \tau_{D_2}\gamma_{D_2}\hat{D}_2. \tag{44}$$

**Step 3.** From the third equation of Equation (28), let $u_f(t) = \dot{u}(t)$, it can be obtained:

$$\begin{aligned} \dot{z}_3 &= \dot{g}_2 - \dot{x}_{2d} \\ &= \tfrac{\partial g_2}{\partial x_1}\dot{x}_1 + \tfrac{\partial g_2}{\partial x_2}\dot{x}_2 + \tfrac{\partial g_2}{\partial u}\dot{u} - \dot{x}_{2d} \\ &= \tfrac{\partial g_2}{\partial x_1}(f_1 + g_1 + d_1 + \Delta f_1) + \tfrac{\partial g_2}{\partial x_2}(f_2 + g_2 + \Delta f_2 + d_2) + \tfrac{\partial g_2}{\partial u}u_f - \dot{x}_{2d}. \end{aligned} \tag{45}$$

The virtual control signal is designed as:

$$u_f = \left(\frac{\partial g_2}{\partial u}\right)^{-1} \begin{pmatrix} -k_3\left(z_3^T z_3\right)^{l-1} z_3 - (\frac{1}{2} + 2\underline{\pi}^2) z_3 - \frac{\partial g_2}{\partial x_1}\left(f_1 + g_1 + \hat{D}_1 \tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right) + \hat{W}_{a1}^T \Phi_{a1}(x_1)\right) \\ -\frac{\partial g_2}{\partial x_2}\left(f_2 + g_2 + \hat{D}_2 \tanh\left(\frac{\partial g_1}{\partial x_2} \cdot \frac{z_2}{\varepsilon_{D_2}}\right) + \hat{W}_{a2}^T \Phi_{a2}(x_1, x_2)\right) + \dot{x}_{2d} \end{pmatrix}. \quad (46)$$

We can obtain:

$$\begin{aligned} z_3^T \dot{z}_3 = & -k_3\left(z_3^T z_3\right)^l + \frac{\partial g_2}{\partial x_1} z_3 \left(d_1 + \varepsilon_{a_1} - \hat{D}_1 \tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right)\right) - \frac{\partial g_2}{\partial x_1} z_3 \widetilde{W}_{a1}^T \Phi_{a1}(x_1) \\ & + \frac{\partial g_2}{\partial x_2} z_3 \left(d_2 + \varepsilon_{a_2} - \hat{D}_2 \tanh\left(\frac{\partial g_1}{\partial x_2} \cdot \frac{z_2}{\varepsilon_{D_2}}\right)\right) - \frac{\partial g_2}{\partial x_2} z_3 \widetilde{W}_{a2}^T \Phi_{a2}(x_1, x_2). \end{aligned} \quad (47)$$

### 3.3. Proof of Stability

**Theorem 1.** *For a second-order non-affine nonlinear system that satisfies Assumptions 1–2 and Lemmas 1–4, if the virtual control signals, adaptive laws, and reinforcement learning (RL) update laws are designed as described in Section 3.1, the closed-loop system can achieve the control the objectives (a) and (b).*

**Proof.** ① Choose the overall candidate Lyapunov function as:

$$\begin{aligned} V = & \tfrac{1}{2} z_1^T z_1 + \tfrac{1}{2} z_2^T z_2 + \tfrac{1}{2} z_3^T z_3 + \tfrac{1}{2} y_1^T y_1 + \tfrac{1}{2} y_2^T y_2 + \tfrac{1}{2} Tr(\widetilde{W}_{a1}^T \eta_1^{-1} \widetilde{W}_{a1}) + \tfrac{1}{2} Tr(\widetilde{W}_{c1}^T \omega_1^{-1} \widetilde{W}_{c1}) \\ & + \tfrac{1}{2} Tr(\widetilde{W}_{a2}^T \eta_2^{-1} \widetilde{W}_{a2}) + \tfrac{1}{2} Tr(\widetilde{W}_{c2}^T \omega_2^{-1} \widetilde{W}_{c2}) + \tfrac{1}{2\tau_{D1}} \widetilde{D}_1^T \widetilde{D}_1 + \tfrac{1}{2\tau_{D2}} \widetilde{D}_2^T \widetilde{D}_2 \end{aligned} \quad (48)$$

By taking its derivative, we can obtain:

$$\begin{aligned} \dot{V} = & z_1^T \dot{z}_1 + z_2^T \dot{z}_2 + z_3^T \dot{z}_3 + y_1^T \dot{y}_1 + y_2^T \dot{y}_2 + \tfrac{1}{\tau_{D_1}} \widetilde{D}_1^T \dot{\hat{D}}_1 + \tfrac{1}{\tau_{D_2}} \widetilde{D}_2^T \dot{\hat{D}}_2 + Tr(\widetilde{W}_{a1}^T \eta_1^{-1} \dot{\hat{W}}_{a1}) \\ & + Tr(\widetilde{W}_{c1}^T \omega_1^{-1} \dot{\hat{W}}_{c1}) + Tr(\widetilde{W}_{a2}^T \eta_2^{-1} \dot{\hat{W}}_{a1}) + Tr(\widetilde{W}_{c2}^T \omega_2^{-1} \dot{\hat{W}}_{c2}) \\ = & -k_1\left(z_1^T z_1\right)^l - k_2\left(z_2^T z_2\right)^l - k_3\left(z_3^T z_3\right)^l - \tfrac{3}{2} z_1^T z_1 - (\tfrac{1}{2} + \tfrac{5}{2}\underline{\pi}^2) z_2^T z_2 - (\tfrac{1}{2} + 2\underline{\pi}^2) z_3^T z_3 - \tfrac{y_1^T y_1}{\delta_1} \\ & - \dot{x}_{1c}^T y_1 - \tfrac{y_2^T y_2}{\delta_2} - \dot{x}_{2c}^T y_2 + z_1(z_2 + y_1) + \frac{\partial g_1}{\partial x_2} z_2 (z_3 + y_2) + \left(d_1 + \varepsilon_{a_1} - \hat{D}_1 \tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right)\right) z_1 \\ & + (\frac{\partial g_1}{\partial x_1} z_2 + \frac{\partial g_2}{\partial x_1} z_3)\left(d_1 + \varepsilon_{a_1} - \hat{D}_1 \tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right)\right) + \left(d_2 + \varepsilon_{a_2} - \hat{D}_2 \tanh\left(\frac{\partial g_1}{\partial x_2} \cdot \frac{z_2}{\varepsilon_{D_2}}\right)\right) \frac{\partial g_1}{\partial x_2} z_2 \\ & + \frac{\partial g_2}{\partial x_2} z_3 \left(d_2 + \varepsilon_{a_2} - \hat{D}_2 \tanh\left(\frac{\partial g_1}{\partial x_2} \cdot \frac{z_2}{\varepsilon_{D_2}}\right)\right) - \widetilde{W}_{a1}^T \Phi_{a1}(x_1)(z_1 + \frac{\partial g_1}{\partial x_1} z_2 + \frac{\partial g_2}{\partial x_1} z_3) \\ & - \widetilde{W}_{a2}^T \Phi_{a2}(x_1, x_2)(\frac{\partial g_1}{\partial x_2} z_2 + \frac{\partial g_2}{\partial x_2} z_3) + \tfrac{1}{\tau_{D_1}} \widetilde{D}_1^T \dot{\hat{D}}_1 + \tfrac{1}{\tau_{D_2}} \widetilde{D}_2^T \dot{\hat{D}}_2 + Tr(\widetilde{W}_{a1}^T \eta_1^{-1} \dot{\hat{W}}_{a1}) \\ & + Tr(\widetilde{W}_{c1}^T \omega_1^{-1} \dot{\hat{W}}_{c1}) + Tr(\widetilde{W}_{a2}^T \eta_2^{-1} \dot{\hat{W}}_{a2}) + Tr(\widetilde{W}_{c2}^T \omega_2^{-1} \dot{\hat{W}}_{c2}). \end{aligned} \quad (49)$$

According to Lemma 3, we have:

$$\begin{aligned} & -\dot{x}_{1c}^T y_1 \le \tfrac{y_1^T y_1}{2\delta_1} + \tfrac{\delta_1 |\dot{x}_{1c}|^2}{2}, \; -\dot{x}_{2c}^T y_2 \le \tfrac{y_2^T y_2}{2\delta_2} + \tfrac{\delta_2 |\dot{x}_{2c}|^2}{2}, \\ & z_1^T(z_2 + y_1) + \frac{\partial g_1}{\partial x_2} z_2 (z_3 + y_2) \le z_1^T z_1 + \left(\tfrac{1}{2} + \underline{\pi}^2\right) z_2^T z_2 + \tfrac{1}{2} z_3^T z_3 + \tfrac{1}{2} y_1^T y_1 + \tfrac{1}{2} y_2^T y_2, \\ & (\frac{\partial g_1}{\partial x_1} z_2 + \frac{\partial g_2}{\partial x_1} z_3)\left(d_1 + \varepsilon_{a_1} - \hat{D}_1 \tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right)\right) \le \tfrac{1}{2}\underline{\pi}^2 z_2^T z_2 + \tfrac{1}{2}\underline{\pi}^2 z_3^T z_3 + \tfrac{3}{2}\bar{\varepsilon}_{D_1}^2, \\ & \frac{\partial g_2}{\partial x_2} z_3 \left(d_2 + \varepsilon_{a_2} - \hat{D}_2 \tanh\left(\frac{\partial g_1}{\partial x_2} \cdot \frac{z_2}{\varepsilon_{D_2}}\right)\right) \le \tfrac{1}{2}\underline{\pi}^2 z_3^T z_3 + \tfrac{1}{2}\bar{\varepsilon}_{D_2}^2, \\ & -\widetilde{W}_{a1}^T \Phi_{a1}(x_1)(z_1 + \tfrac{\partial g_1}{\partial x_1} z_2 + \tfrac{\partial g_2}{\partial x_1} z_3) \le \tfrac{1}{2} z_1^T z_1 + \tfrac{1}{2}\underline{\pi}^2 (z_2^T z_2 + z_3^T z_3) + \tfrac{3}{2}\Phi_{a1M}^2 Tr\left(\widetilde{W}_{a1}{}^T \widetilde{W}_{a1}\right), \\ & -\widetilde{W}_{a2}^T \Phi_{a2}(x_1, x_2)(\tfrac{\partial g_1}{\partial x_2} z_2 + \tfrac{\partial g_2}{\partial x_2} z_3) \le \tfrac{1}{2}\underline{\pi}^2 (z_2^T z_2 + z_3^T z_3) + \Phi_{a2M}^2 Tr\left(\widetilde{W}_{a2}{}^T \widetilde{W}_{a2}\right). \end{aligned} \quad (50)$$

It follows from Lemma 1 that:

$$\begin{aligned} & z_1 \left(d_1 + \varepsilon_{a_1} - \hat{D}_1 \tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right)\right) \le -\widetilde{D}_1 z_1 \tanh\left(\frac{z_1}{\varepsilon_{D_1}}\right) + \kappa \varepsilon_{D_1}, \\ & \frac{\partial g_1}{\partial x_2} z_2 \left(d_2 + \varepsilon_{a_2} - \hat{D}_2 \tanh\left(\frac{\partial g_1}{\partial x_2} \cdot \frac{z_2}{\varepsilon_{D_2}}\right)\right) \le -\widetilde{D}_2 \frac{\partial g_1}{\partial x_2} z_2 \tanh\left(\frac{\partial g_1}{\partial x_2} \cdot \frac{z_2}{\varepsilon_{D_2}}\right) + \kappa \varepsilon_{D_2}. \end{aligned} \quad (51)$$

Combing Equations (38) and (44), we can obtain:

$$\begin{aligned}&\tfrac{1}{\tau_{D_1}}\widetilde{D}_1^T\dot{\widetilde{D}}_1 = \widetilde{D}_1 z_1 \tanh\left(\tfrac{z_1}{\varepsilon_{D_1}}\right) - \gamma_{D_1}\hat{D}_1\widetilde{D}_1,\\&\tfrac{1}{\tau_{D_2}}\widetilde{D}_2^T\dot{\widetilde{D}}_2 = \widetilde{D}_2\tfrac{\partial g_1}{\partial x_2} z_2 \tanh\left(\tfrac{\partial g_1}{\partial x_2}\cdot\tfrac{z_2}{\varepsilon_{D_2}}\right) - \gamma_{D_2}\hat{D}_2\widetilde{D}_2.\end{aligned} \tag{52}$$

Through transforming the inequalities, we derive the following relationship:

$$\begin{aligned}&-\gamma_{D_1}\hat{D}_1^T\widetilde{D}_1 \le -\tfrac{1}{2}\gamma_{D_1}\widetilde{D}_1^T\widetilde{D}_1 + \tfrac{1}{2}\gamma_{D_1}D_1^TD_1,\\&-\gamma_{D_2}\hat{D}_2^T\widetilde{D}_2 \le -\tfrac{1}{2}\gamma_{D_2}\widetilde{D}_2^T\widetilde{D}_2 + \tfrac{1}{2}\gamma_{D_2}D_2^TD_2,\\&Tr(\widetilde{W}_{a1}^T\eta_1^{-1}\dot{\widetilde{W}}_{a1}) + Tr(\widetilde{W}_{c1}^T\omega_1^{-1}\dot{\widetilde{W}}_{c1})\\&\le -\tfrac{1}{2}(\tau_1 - \Phi_{a1M}^2)Tr\left(\widetilde{W}_{a1}{}^T\widetilde{W}_{a1}\right) - \tfrac{1}{2}(\omega_1 - \Phi_{c1m}^2 - \Phi_{c1M}^2\Gamma_{1M}^2)Tr\left(\widetilde{W}_{c1}{}^T\widetilde{W}_{c1}\right)\\&\quad + \tfrac{1}{2}(\tau_1 + \Phi_{a1M}^2)Tr(W_{a1}{}^TW_{a1}) + \tfrac{1}{2}(\omega_1 + 2\Phi_{c1m}^2 + \Phi_{c1M}^2\Gamma_{1M}^2)Tr(W_{c1}^TW_{c1}) + \tfrac{1}{2}\varepsilon_{c1m}^2,\\&Tr(\widetilde{W}_{a2}^T\eta_2^{-1}\dot{\widetilde{W}}_{a2}) + Tr(\widetilde{W}_{c2}^T\omega_2^{-1}\dot{\widetilde{W}}_{c2})\\&\le -\tfrac{1}{2}(\tau_2 - \underline{\pi}^2\Phi_{a2M}^2 - 2\pi\Phi_{a2M}^2)Tr(\widetilde{W}_{a2}{}^T\widetilde{W}_{a2}) - \tfrac{1}{2}(\omega_2 - \Phi_{c2m}^2 - \underline{\pi}\Phi_{c2M}^2\Gamma_{2M}^2)Tr(\widetilde{W}_{c2}^T\widetilde{W}_{c2})\\&+\tfrac{1}{2}(\underline{\pi}^2\Phi_{a2M}^2 + \tau_2)Tr(W_{a2}{}^TW_{a2}) + \tfrac{1}{2}(\underline{\pi}\Phi_{c2M}^2\Gamma_{2M}^2 + \omega_2 + 2\Phi_{c2m}^2)Tr(W_{c2}^TW_{c2}) + \tfrac{1}{2}\varepsilon_{c2m}^2.\end{aligned} \tag{53}$$

It can be sorted out:

$$\begin{aligned}&\dot{V} \le -k_1(z_1^Tz_1)^l - k_2(z_2^Tz_2)^l - k_3(z_3^Tz_3)^l - (\tfrac{1}{2\delta_1} - \tfrac{1}{2})y_1^Ty_1 - (\tfrac{1}{2\delta_2} - \tfrac{1}{2})y_2^Ty_2\\&-\tfrac{1}{2}\gamma_{D_1}\widetilde{D}_1^T\widetilde{D}_1 - \tfrac{1}{2}\gamma_{D_2}\widetilde{D}_2^T\widetilde{D}_2 - \tfrac{1}{2}(\tau_1 - 4\Phi_{a1M}^2)Tr\left(\widetilde{W}_{a1}{}^T\widetilde{W}_{a1}\right) - \tfrac{1}{2}(\omega_1 - \Phi_{c1m}^2 - \Phi_{c1M}^2\Gamma_{1M}^2)Tr\left(\widetilde{W}_{c1}{}^T\widetilde{W}_{c1}\right)\\&-\tfrac{1}{2}(\tau_2 - \underline{\pi}^2\Phi_{a2M}^2 - 2\underline{\pi}\Phi_{a2M}^2 - 2\Phi_{a2M}^2)Tr(\widetilde{W}_{a2}{}^T\widetilde{W}_{a2}) - \tfrac{1}{2}(\omega_2 - \Phi_{c2m}^2 - \underline{\pi}\Phi_{c2M}^2\Gamma_{2M}^2)Tr(\widetilde{W}_{c2}^T\widetilde{W}_{c2})\\&+\kappa\varepsilon_{D_1} + \tfrac{3}{2}\bar{\varepsilon}_{D_1}^2 + \kappa\varepsilon_{D_2} + \tfrac{1}{2}\bar{\varepsilon}_{D_2}^2 + \tfrac{1}{2}\gamma_{D_1}D_1^TD_1 + \tfrac{1}{2}\gamma_{D_2}D_2^TD_2 + \tfrac{\delta_1|\dot{x}_{1c}|^2}{2} + \tfrac{\delta_2|\dot{x}_{2c}|^2}{2}\\&+\tfrac{1}{2}(\tau_1 + \Phi_{a1M}^2)Tr(W_{a1}{}^TW_{a1}) + \tfrac{1}{2}(\omega_1 + 2\Phi_{c1m}^2 + \Phi_{c1M}^2\Gamma_{1M}^2)Tr(W_{c1}^TW_{c1}) + \tfrac{1}{2}\varepsilon_{c1m}^2\\&+\tfrac{1}{2}(\underline{\pi}^2\Phi_{a2M}^2 + \tau_2)Tr(W_{a2}{}^TW_{a2}) + \tfrac{1}{2}(\underline{\pi}\Phi_{c2M}^2\Gamma_{2M}^2 + \omega_2 + 2\Phi_{c2m}^2)Tr(W_{c2}^TW_{c2}) + \tfrac{1}{2}\varepsilon_{c2m}^2.\end{aligned} \tag{54}$$

By defining:

$$\overline{C} = \min\left\{\begin{array}{l}2k_1, 2k_2, 2k_3, (\tfrac{1}{\delta_1} - 1), (\tfrac{1}{\delta_2} - 1), \tau_{D_1}\gamma_{D_1}, \tau_{D_2}\gamma_{D_2}, (\tau_1 - 4\Phi_{a1M}^2), (\omega_1 - \Phi_{c1m}^2 - \Phi_{c1M}^2\Gamma_{1M}^2),\\(\tau_2 - \underline{\pi}^2\Phi_{a2M}^2 - 2\underline{\pi}\Phi_{a2M}^2 - 2\Phi_{a2M}^2), (\omega_2 - \Phi_{c2m}^2 - \underline{\pi}\Phi_{c2M}^2\Gamma_{2M}^2)\end{array}\right\}, \tag{55}$$

$$\begin{aligned}\varepsilon &= \kappa\varepsilon_{D_1} + \tfrac{3}{2}\bar{\varepsilon}_{D_1}^2 + \kappa\varepsilon_{D_2} + \tfrac{1}{2}\bar{\varepsilon}_{D_2}^2 + \tfrac{1}{2}\gamma_{D_1}D_1^TD_1 + \tfrac{1}{2}\gamma_{D_2}D_2^TD_2 + \tfrac{\delta_1|\dot{x}_{1c}|^2}{2} + \tfrac{\delta_2|\dot{x}_{2c}|^2}{2}\\&+\tfrac{1}{2}(\tau_1 + \Phi_{a1M}^2)Tr(W_{a1}{}^TW_{a1}) + \tfrac{1}{2}(\omega_1 + 2\Phi_{c1m}^2 + \Phi_{c1M}^2\Gamma_{1M}^2)Tr(W_{c1}^TW_{c1}) + \tfrac{1}{2}\varepsilon_{c1m}^2\\&+\tfrac{1}{2}(\underline{\pi}^2\Phi_{a2M}^2 + \tau_2)Tr(W_{a2}{}^TW_{a2}) + \tfrac{1}{2}(\underline{\pi}\Phi_{c2M}^2\Gamma_{2M}^2 + \omega_2 + 2\Phi_{c2m}^2)Tr(W_{c2}^TW_{c2}) + \tfrac{1}{2}\varepsilon_{c2m}^2.\end{aligned} \tag{56}$$

Then, Equation (54) can be rewritten as:

$$\begin{aligned}&\dot{V} \le -\tfrac{1}{2}\overline{C}(z_1^Tz_1)^l - \tfrac{1}{2}\overline{C}(z_2^Tz_2)^l - \tfrac{1}{2}\overline{C}(z_3^Tz_3)^l - \tfrac{1}{2}\overline{C}y_1^Ty_1 - \tfrac{1}{2}\overline{C}y_2^Ty_2\\&-\tfrac{\overline{C}}{2\tau_{D1}}\widetilde{D}_1^T\widetilde{D}_1 - \tfrac{\overline{C}}{2\tau_{D2}}\widetilde{D}_2^T\widetilde{D}_2 - \tfrac{\overline{C}}{2\eta_1}Tr\left(\widetilde{W}_{a1}{}^T\widetilde{W}_{a1}\right) - \tfrac{\overline{C}}{2\omega_1}Tr\left(\widetilde{W}_{c1}{}^T\widetilde{W}_{c1}\right)\\&-\tfrac{\overline{C}}{2\eta_1}Tr(\widetilde{W}_{a2}{}^T\widetilde{W}_{a2}) - \tfrac{\overline{C}}{2\omega_2}Tr(\widetilde{W}_{c2}^T\widetilde{W}_{c2}) + \varepsilon.\end{aligned} \tag{57}$$

According to Lemma, we obtain:

$$\begin{aligned}&\left(\tfrac{1}{2}y_i^Ty_i\right)^l \le (1-l)l^{\frac{l}{1-l}} + \tfrac{1}{2}y_i^Ty_i,\\&\left(\tfrac{1}{2\tau_{Di}}\widetilde{D}_i^T\widetilde{D}_i\right)^l \le (1-l)l^{\frac{l}{1-l}} + \tfrac{1}{2\tau_{Di}}\widetilde{D}_i^T\widetilde{D}_i,\\&\left(\tfrac{1}{2\eta_i}Tr\left(\widetilde{W}_{ai}{}^T\widetilde{W}_{ai}\right)\right)^l \le (1-l)l^{\frac{l}{1-l}} + \tfrac{1}{2\eta_i}Tr\left(\widetilde{W}_{ai}{}^T\widetilde{W}_{ai}\right),\\&\left(\tfrac{1}{2\omega_i}Tr\left(\widetilde{W}_{ci}{}^T\widetilde{W}_{ci}\right)\right)^l \le (1-l)l^{\frac{l}{1-l}} + \tfrac{1}{2\omega_i}Tr\left(\widetilde{W}_{ci}{}^T\widetilde{W}_{ci}\right).\end{aligned} \tag{58}$$

It follows from Lemma 4 that:

$$-\overline{C}\left(\frac{1}{2}\left(z_i^Tz_i\right)^l\right) \le -2^{l-1}\overline{C}\left(\frac{1}{2}z_i^Tz_i\right)^l. \tag{59}$$

Finally, we can obtain:

$$\begin{aligned}\dot{V} \leq & -2^{l-1}\overline{C}\left(\tfrac{1}{2}z_1^T z_1\right)^l - 2^{l-1}\overline{C}\left(\tfrac{1}{2}z_2^T z_2\right)^l - 2^{l-1}\overline{C}\left(\tfrac{1}{2}z_3^T z_3\right)^l - \overline{C}\left(\tfrac{1}{2}y_1^T y_1\right)^l + 9\overline{C}(1-l)l^{\frac{l}{1-l}} \\ & -\overline{C}\left(\tfrac{1}{2}y_2^T y_2\right)^l - \overline{C}\left(\tfrac{1}{2}y_3^T y_3\right)^l - \overline{C}\left(\tfrac{1}{2\tau_{D1}}\widetilde{D}_1^T\widetilde{D}_1\right)^l - \overline{C}\left(\tfrac{1}{2\tau_{D2}}\widetilde{D}_2^T\widetilde{D}_2\right)^l \\ & -\overline{C}\left(\tfrac{1}{2\eta_1}Tr\left(\widetilde{W}_{a1}{}^T\widetilde{W}_{a1}\right)\right)^l - \overline{C}\left(\tfrac{1}{2\eta_2}Tr\left(\widetilde{W}_{a2}{}^T\widetilde{W}_{a2}\right)\right) \\ & -\left(\tfrac{1}{2\varpi_2}Tr\left(\widetilde{W}_{c2}{}^T\widetilde{W}_{c2}\right)\right)^l + \varepsilon.\end{aligned} \tag{60}$$

Let:

$$\begin{aligned}C &= \min\left\{2^{l-1}\overline{C}, \overline{C}\right\} \\ D &= \varepsilon + 9\overline{C}(1-l)l^{\frac{l}{1-l}}.\end{aligned} \tag{61}$$

Then, Equation (60) can be rewritten as:

$$\dot{V} \leq -CV^l + D. \tag{62}$$

Based on Lemma 2, selecting the parameters $k_i > 0, i = 1,2,3, 0 < \delta_i < 1, \tau_{D_i} > 0, \gamma_{D_i} > 0, \tau_1 > 4\Phi_{a1M}^2, \omega_1 > \Phi_{c1m}^2 + \Phi_{c1M}^2\Gamma_{1M}^2, \tau_2 > \underline{\pi}^2\Phi_{a2M}^2 + 2\underline{\pi}\Phi_{a2M}^2 + 2\Phi_{a2M}^2, \omega_2 > \Phi_{c2m}^2 + \underline{\pi}\Phi_{c2M}^2\Gamma_{2M}^2 + \underline{\pi}\Phi_{c2M}^2\Gamma_{2M}^2, i = 1,2$ to derive the explicit conditions for $\overline{C} > 0$, we conclude that all the signals of the closed-loop system are semi-global practical finite-time stable.

② Given $T_r = \frac{1}{(1-l)\aleph C}\left[V^{1-l}(0) - \left(\frac{D}{(1-\aleph)C}\right)^{\Xi}\right]$, where $\Xi = (1-l)/l, \aleph \in (0,1)$. With the analysis in [52], it yields:

$$V \leq \left(\frac{D}{(1-\aleph)C}\right)^{1/l}, \forall t \geq T_r. \tag{63}$$

Then, combining Equation (48), it can be obtained:

$$|z_1(t)| = |x_1(t) - y_d| < 2\left(\frac{D}{(1-\aleph)C}\right)^{\frac{1}{2l}}, \forall t \geq T_r. \tag{64}$$

Thereby, it can be concluded that the tracking error converges to and remains in a small neighborhood of the desired signals for $\forall t \geq T_r$. □

**Remark 7.** *The system tracking error $z_1(t)$enters and remains within a steady-state neighborhood determined by the parameters $D, C, \aleph, l$, after a finite time $T_r$. The convergence process can be accelerated by appropriately increasing the value of $l$.*

## 4. Simulation Study

In this section, numerical simulations are performed to confirm the effectiveness and advantages of the developed RL-based finite-time adaptive fault tolerant control method for a morphing unmanned aircraft.

The initial conditions are chosen as: $\begin{bmatrix}\alpha & \beta & \gamma_v\end{bmatrix}^T = \begin{bmatrix}20^\circ & 5^\circ & -5^\circ\end{bmatrix}^T$, $\begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T = \begin{bmatrix}0.1\ rad/s & 0.1\ rad/s & 0.1\ rad/s\end{bmatrix}^T$; The adaptive parameters and RL parameters are set as: $k_1 = 3$, $k_2 = 5$, $k_3 = 8$, $\tau_1 = \tau_2 = 8$, $\eta_1 = 10$, $\eta_2 = 1$, $\varpi_1 = 5$, $\varpi_2 = 0.5$, $\omega_1 = 0.5$, $\omega_2 = 0.1$; The disturbance compensation adaptative parameters are defined as: $\varepsilon_{D_1} = \varepsilon_{D_2} = 0.01$, $\tau_{D_1} = \tau_{D_2} = 10$, $\gamma_{D_1} = \gamma_{D_2} = 10$; Structural parameters of the morphing unmanned aircraft: $I_{bx} = 20$, $I_{by} = I_{bz} = 150$; The aircraft's flight velocity is $V = 600\ m/s$, and its flight altitude is $H = 15$ km. To ensure the proper implementation of

the simulations, they are conducted under three different cases. The different parameters of the uncertainties $\Delta f_1, \Delta f_2$ and disturbances $d_1(t)$ and $d_2(t)$ expressed in model (27) are defined under three cases. The detailed specifications of these scenarios are presented in Table 1.

**Table 1.** The different parameter values of the morphing unmanned aircraft under three cases.

| Cases | $\Delta f_1$ | $d_1(t)$ | $\Delta f_2$ | $d_2(t)$ |
|---|---|---|---|---|
| Case 1 | $\dot{\theta} = 0.007^\circ/s$ | $\dot{\sigma} = 0.01^\circ/s, \theta = 0^\circ$ | $0.01 I_b$ | $0.01 M_f + 0.01 I_b$ |
| Case 2 | $\dot{\theta} = 0.01^\circ/s$ | $\dot{\sigma} = 0.006^\circ/s, \theta = 0.03^\circ$ | $0.008 I_b$ | $0.006 M_f + 0.01 I_b$ |
| Case 3 | $\dot{\theta} = 0.005^\circ \times (1 + 0.1\sin(t))/s$ | $\dot{\sigma} = 0.008^\circ \times (1 + 0.1\cos(t))/s$, $\theta = 0.02^\circ$ | $0.004 \times (1 + 0.1\sin(t)) I_b$ | $0.009 M_f + 0.01 \times (1 + 0.1\cos(t)) I_f$ |

The simulation results are given in Figures 2–6. Figure 2 demonstrates the trajectory tracking performance of the morphing unmanned aircraft's attitude angles under coupled uncertainties. Obviously, the proposed RL-based adaptive controller achieves convergence within a 0.001-error bound around the desired trajectories. Despite an initial transient overshoot, the intelligent controller can quickly compensate for the uncertainties, exhibiting strong adaptability and robustness. The disturbance estimation and neural network weight estimation signals are shown in Figures 3–5, and it can be seen that all signals are bounded. Figure 6 shows the control torque variation curve over time under Case 1.

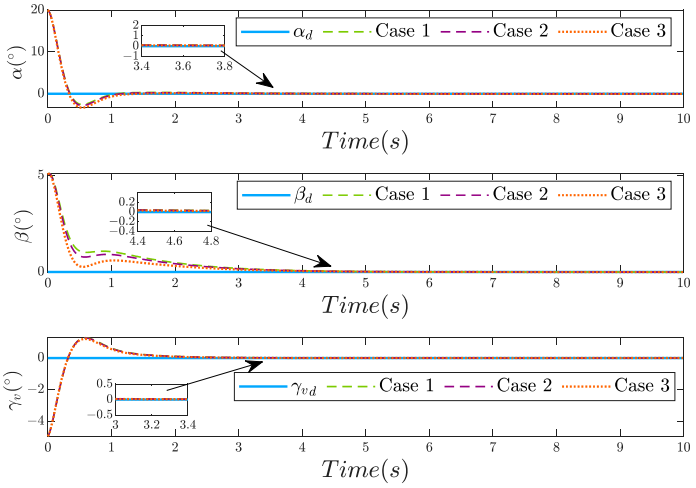



**Figure 2.** Attitude angle trajectories under three cases.

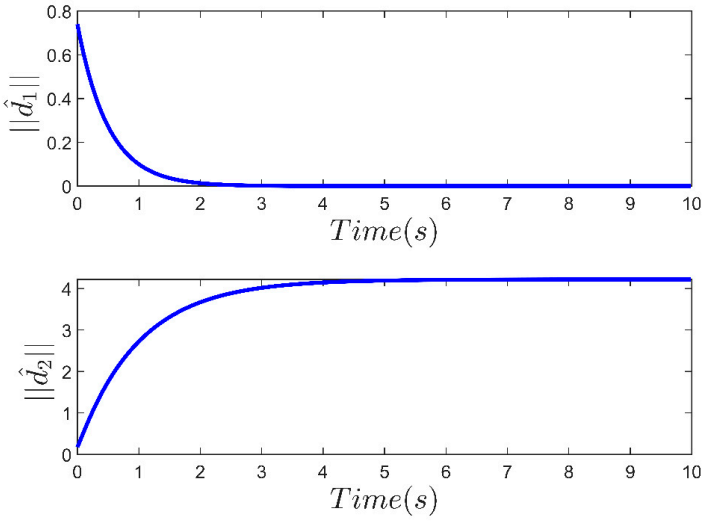



**Figure 3.** Disturbance norm estimation under Case 1.

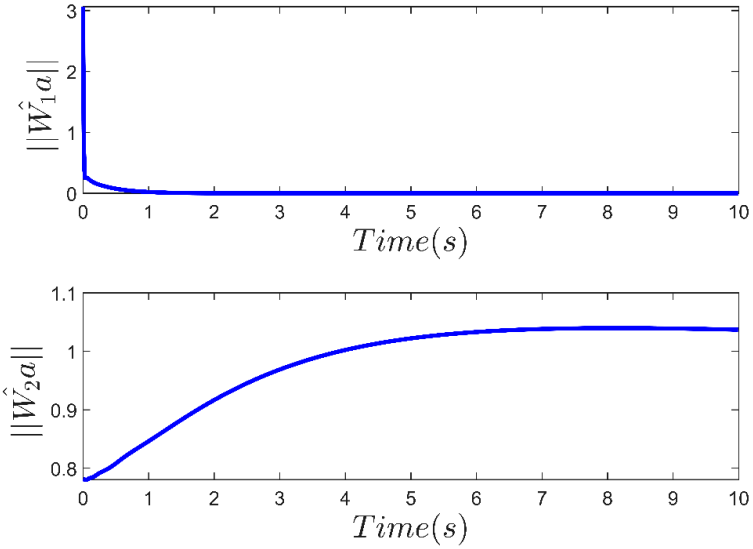



**Figure 4.** Actor network weight norm estimation under Case 1.

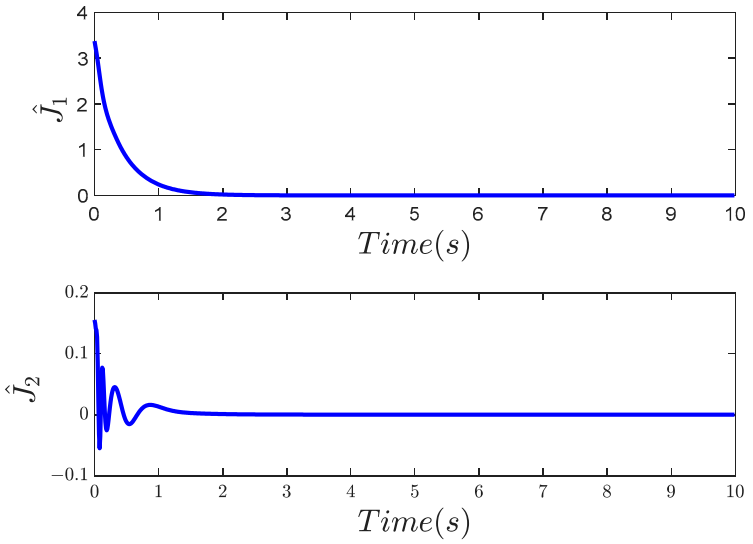



**Figure 5.** Penalty functions estimation under Case 1.

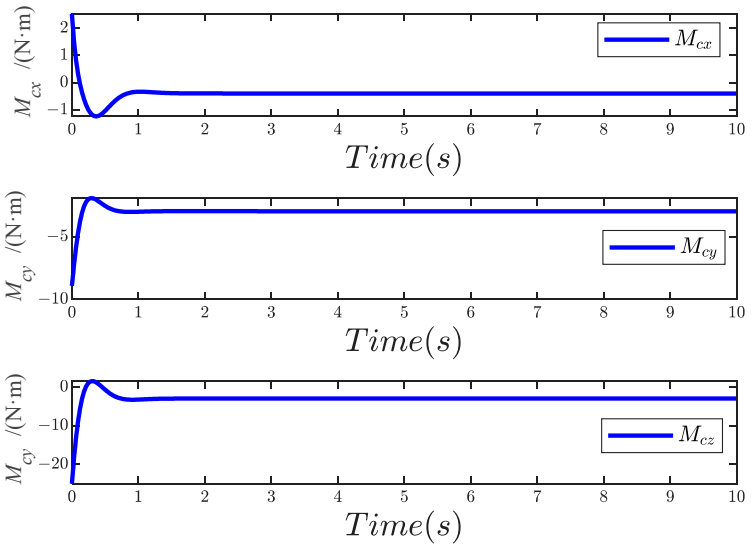



**Figure 6.** Control torques under Case 1.

To verify the advantages of the proposed reinforcement learning adaptive control (RLAC) method and its adaptability, such as speed and stability, the neural network adaptive control (NNAC) [59] method and the adaptive fault-tolerant control (AFTC) [60] method are selected for comparison. These existing adaptive control methods serve as benchmarks to highlight the unique features of the RLAC method. All controllers track identical reference trajectories under the same experimental conditions. The experimental results comprehensively compare the proposed RLAC method with two existing control approaches, AFTC and NNAC methods, are shown in Figures 7 and 8. Table 2 presents the steady-state error and convergence time of the three methods, where convergence time is defined as the time required to complete 90% of the convergence process. In Figure 8, the tracking error is defined as the sum of the two norms of the errors $z_1$ and $z_2$, that is, $e = \|z_1\|_2 + \|z_2\|_2$, which clearly highlights the superiority of the proposed RLAC method.

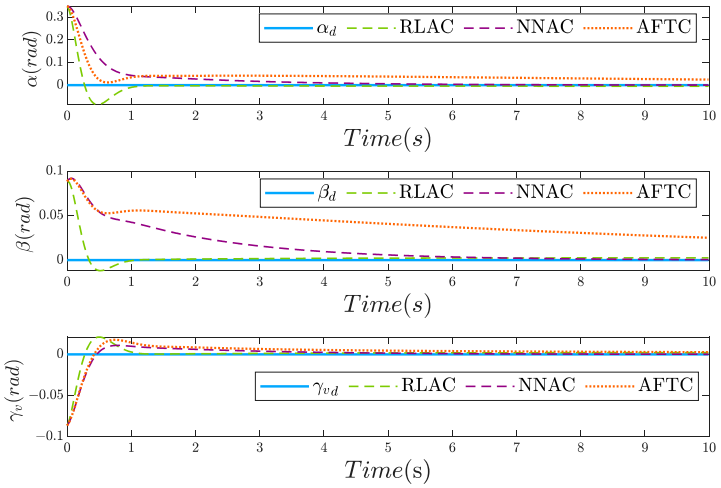



**Figure 7.** Comparison of tracking performance of three methods under Case 1.

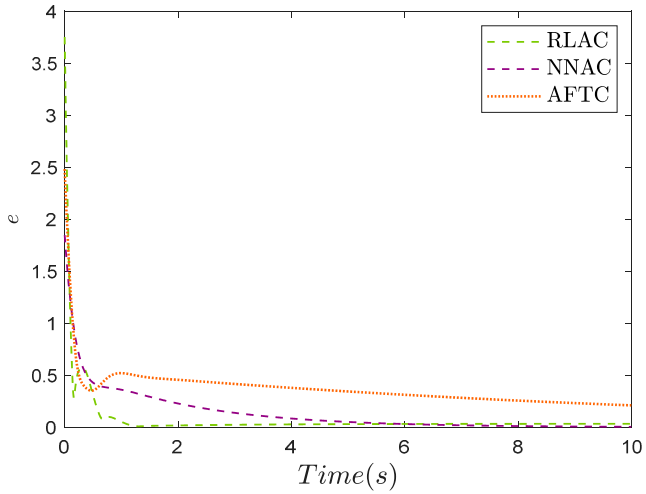



**Figure 8.** Comparison of tracking errors of three methods under Case 1.

**Table 2.** Tracking performance of three methods.

| Method | Steady-State Error (Rad) | Convergence Time (s) |
|---|---|---|
| RLAC | 0.0038, 0.0024, 0.00087 | 0.79, 0.66, 0.82 |
| NNAC | 0.007, 0.0023, 0.001 | 1.43, 4.19, 1.19 |
| AFTC | >0.026, >0.025, >0.0026 | 5.81, >10, 1.9 |

The results demonstrate that under the same experimental conditions, the AFTC method has limitations. It has a relatively large steady-state error of about 0.25 and a convergence time exceeding 10 s, which means it struggles to precisely track the reference trajectory and reaches a stable state slowly, leading to inefficient system operation.

The NNAC method shows improvements. It can achieve a steady-state error of less than 0.0015, more accurate than the AFTC method. However, it suffers from significant overshoot, which may cause system instability. In contrast, the RLAC method demonstrates remarkable superiority. It has a steady-state error below 0.001, the fastest convergence speed, and can rapidly adapt to changing environments. In terms of tracking performance, it outperforms both AFTC and NNAC methods in speed and stability. Consequently, the RLAC method offers significant advantages over AFTC and NNAC methods, making it a more promising choice for practical control applications.

## 5. Conclusions

The RL-based adaptive fault-tolerant control method proposed in this paper effectively solves the control problem for a class of morphing unmanned aircraft under mismatched disturbances and coupled uncertainties. The aircraft's uncertainties are modeled as a non-affine second-order nonlinear system, and the non-affine structure is handled by introducing an auxiliary integral system. The unknown functions are evaluated by the introduced RL algorithms, and the control actions are adjusted by the developed RL-based adaptive laws. The filtering errors and disturbances are compensated by adopting a disturbance boundary estimator. The RL-based adaptive fault-tolerant control method has been successfully designed by integrating RL, disturbance estimation, and finite-time theory. It is proven by the Lyapunov function that the system has finite-time stability and all signals are bounded. Numerical simulations verify the effectiveness and superiority of this method. In the future, this method is expected to be extended to aircraft models, considering various factors in the actual flight environment.

**Author Contributions:** W.R.: writing–review and editing, writing–original draft, visualization, methodology, software, conceptualization, and funding acquisition. Y.W.: writing—review and editing, supervision. C.W.: writing—review and editing, visualization. Z.W.: writing—review and editing, visualization. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported in part by the Guizhou Provincial Science and Technology Projects ([2025] 049).

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

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

## References

1. Jha, A.K.; Kudva, J.N. Morphing Unmanned aircraft Concepts, Classifications, and Challenges. In *Smart Structures and Materials 2004: Industrial and Commercial Applications of Smart Structures Technologies*; Society of Photo Optical: Bellingham, WA, USA, 2004; Volume 5388, pp. 213–224.
2. Chu, L.; Li, Q.; Gu, F.; Du, X.; He, Y.; Deng, Y. Design, Modeling, and Control of Morphing Aircraft: A Review. *Chin. J. Aeronaut.* **2022**, *35*, 220–246. [CrossRef]
3. Yu, Z.; Zang, Y.; Jiang, B. PID-type fault-tolerant prescribed performance control of fixed-wing UAV. *J. Syst. Eng. Electron.* **2021**, *32*, 1053–1061.
4. Noordin, A.; Mohd Basri, M.A.; Mohamed, Z.; Mat Lazim, I. Adaptive PID controller using sliding mode control approaches for quadrotor UAV attitude and position stabilization. *Arab. J. Sci. Eng.* **2021**, *46*, 963–981. [CrossRef]
5. Wang, P.; Chen, H.; Bao, C.; Tang, G. Review on Modeling and Control Methods of Morphing Vehicle. *J. Astronaut.* **2022**, *43*, 853–865.

6. Ameduri, S.; Concilio, A. Morphing Wings Review: Aims, Challenges, and Current Open Issues of a Technology. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2023**, *237*, 4112–4130. [CrossRef]
7. Shardul, G. Study of Various Trends for Morphing Wing Technology. *J. Comput. Methods Sci. Eng.* **2021**, *21*, 613–621. [CrossRef]
8. He, H.; Wang, P. Integrated Guidance and Control Method for High-Speed Morphing Wing Aircraft. *Acta Aeronaut. Astronaut. Sin.* **2024**, *45*, 299–312.
9. Zhang, H.; Wang, P.; Tang, G.; Bao, W. Fixed-Time Sliding Mode Control for Hypersonic Morphing Vehicles via Event-Triggering Mechanism. *Aerosp. Sci. Technol.* **2023**, *140*, 108458. [CrossRef]
10. Abouheaf, M.; Mailhot, N.Q.; Gueaieb, W.; Spinello, D. Guidance Mechanism for Flexible-Wing Aircraft Using Measurement-Interfaced Machine-Learning Platform. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 4637–4648. [CrossRef]
11. Lee, J.; Kim, Y. Neural Network-Based Nonlinear Dynamic Inversion Control of Variable-Span Morphing Aircraft. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2020**, *234*, 1624–1637. [CrossRef]
12. Irfan, S.; Zhao, L.; Ullah, S.; Javaid, U.; Iqbal, S. Differentiator- and Observer-Based Feedback Linearized Advanced Nonlinear Control Strategies for an Unmanned Aerial Vehicle System. *Drones* **2024**, *8*, 527. [CrossRef]
13. Lee, H.; Kim, S.; Kim, Y. Actor-Critic-Based Optimal Adaptive Control Design for Morphing Aircraft. *IFAC Pap.* **2020**, *53*, 14863–14868. [CrossRef]
14. Zhou, Y.; Wang, P.; Tang, G.; Chen, H. Disturbance Observer-Based Prescribed Performance Control for Morphing Aircraft. *Tactical Missile Technol.* **2024**, *4*, 72–82.
15. Hu, H.; Li, Y.; Yi, W.; Wang, Y.; Qu, F.; Wang, X. Event-Triggered Neural Network-Based Adaptive Control for a Class of Uncertain Nonlinear Systems. *J. Circuits Syst. Comput.* **2021**, *30*, 15. [CrossRef]
16. Yuan, F.; Liu, Y.-J.; Liu, L.; Lan, J. Adaptive Neural Network Control of Non-Affine Multi-Agent Systems with Actuator Fault and Input Saturation. *Int. J. Robust. Nonlinear Control* **2024**, *34*, 3761–3780. [CrossRef]
17. Anderson, R.B.; Marshall, J.A.; L'Afflitto, A.; Dotterweich, J.M. Model Reference Adaptive Control of Switched Dynamical Systems with Applications to Aerial Robotics. *J. Intell. Robot. Syst.* **2020**, *100*, 1265–1281. [CrossRef]
18. Qi, W.; Teng, J.; Cao, J.; Yan, H.; Cheng, J. Improved Model Reference-Based Adaptive Nonlinear Dynamic Inversion for Fault-Tolerant Flight Control. *Int. J. Robust. Nonlinear Control* **2023**, *33*, 10328–10359. [CrossRef]
19. Li, Y.; Liu, X.; Ming, R.; Li, K.; Zhang, W. Dynamic Protocol-Based Control for Hidden Stochastic Jump Multiarea Power Systems in Finite-Time Interval. *IEEE Trans. Cybern.* **2025**, *55*, 1486–1496.
20. Li, G.; Peng, C.; Cao, Z. Finite-time bounded asynchronous sliding-mode control for T-S fuzzy time-delay systems via event-triggered scheme. *Fuzzy Sets Syst.* **2025**, *514*, 109400. [CrossRef]
21. Yu, C.; Jiang, J.; Wang, S.; Han, B. Fixed-Time Adaptive General Type-2 Fuzzy Logic Control for Air-Breathing Hypersonic Vehicle. *Trans. Inst. Meas. Control* **2021**, *43*, 2143–2158. [CrossRef]
22. Hernández-González, O.; Targui, B.; Valencia-Palomo, G.; Guerrero-Sánchez, M.E. Robust cascade observer for a disturbance unmanned aerial vehicle carrying a load under multiple time-varying delays and uncertainties. *Int. J. Syst. Sci.* **2024**, *55*, 1056–1072. [CrossRef]
23. Hernández-González, O.; Ramírez-Rasgado, F.; Farza, M.; Guerrero-Sánchez, M.-E.; Astorga-Zaragoza, C.-M.; M'Saad, M.; Valencia-Palomo, G. Observer for Nonlinear Systems with Time-Varying Delays: Application to a Two-Degrees-of-Freedom Helicopter. *Aerospace* **2024**, *11*, 206. [CrossRef]
24. Qu, C.; Cheng, L.; Gong, S.; Huang, X. Dynamic-Matching Adaptive Sliding Mode Control for Hypersonic Vehicles. *Aerosp. Sci. Technol.* **2024**, *149*, 109159. [CrossRef]
25. Zhao, Y.; Ma, Y. Adaptive Event-Triggered Finite-Time Sliding Mode Control for Singular T–S Fuzzy Markov Jump Systems with Asynchronous Modes. *Commun. Nonlinear Sci. Numer. Simul.* **2023**, *126*, 107465. [CrossRef]
26. Shi, X.; Li, Y.; Liu, Q.; Lin, K.; Chen, S. A Fully Distributed Adaptive Event-Triggered Control for Output Regulation of Multi-Agent Systems with Directed Network. *Inf. Sci.* **2023**, *626*, 60–74. [CrossRef]
27. Abbas, M.; Sadati, S.H.; Khazaee, M. Fault-Tolerant Control Design Based on Observer-Switching and Adaptive Neural Networks for Maneuvering Aircraft. *J. Braz. Soc. Mech. Sci. Eng.* **2024**, *46*, 14689–14698. [CrossRef]
28. Zhao, L.; Wang, L.; Cao, Y.; Yang, Y.; Wen, S. Learning-Based Fault-Tolerant Control With High-Order Control Barrier Functions. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 694. [CrossRef]
29. Ma, J.; Peng, C. Adaptive Model-Free Fault-Tolerant Control Based on Integral Reinforcement Learning for a Highly Flexible Aircraft with Actuator Faults. *Aerosp. Sci. Technol.* **2021**, *119*, 107204. [CrossRef]
30. Liu, Y.; Jing, Y.; Liu, X.; Li, X. Survey on Finite-Time Control for Nonlinear Systems. *Control Theory Appl.* **2020**, *37*, 1–12.
31. Guo, F.; Zhang, W.; Lv, M.; Zhang, R. Fault-Tolerant Tracking Control of Hypersonic Vehicle Based on a Universal Prescribe Time Architecture. *Drones* **2024**, *8*, 295. [CrossRef]
32. Bingö, Z.; Güzey, H. Finite-Time Neuro-Sliding-Mode Controller Design for Quadrotor UAVs Carrying Suspended Payload. *Drones* **2022**, *6*, 311. [CrossRef]
33. Rang, E.R. Isochrone Families for Second-Order Systems. *IEEE Trans. Automat. Contr.* **1963**, *8*, 64–65. [CrossRef]

34. Chen, X.; Zhang, X. Output-Feedback Control Strategies of Lower-Triangular Nonlinear Nonholonomic Systems in Any Prescribed Finite Time. *Int. J. Robust Nonlinear Control* **2019**, *29*, 904–918. [CrossRef]
35. Zhou, T.; Liu, S.; Liu, C. Finite-Time Prescribed Performance Adaptive Fuzzy Control for Nonlinear Systems with Unknown Virtual Control Coefficients. In Proceedings of the 2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Chengdu, China, 18–20 June 2021; pp. 7–12.
36. Kang, B.; Li, Y. Adaptive Finite-Time Stabilization of High-Order Stochastic Nonlinear Systems with Unknown Control Direction. In Proceedings of the Conference Digest—2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Chengdu, China, 18–20 June 2021; pp. 150–155.
37. Zhu, W.; Zong, Q.; Zhang, X.; Liu, W. Disturbance Observer-Based Multivariable Finite-Time Attitude Tracking for Flexible Spacecraft. In Proceedings of the Chinese Control Conference (CCC), Chongqing, China, 28–30 July 2020; Volume 2020, pp. 1772–1777.
38. Wang, H.; Kang, S.; Feng, Z. Finite-Time Adaptive Fuzzy Command Filtered Backstepping Control for a Class of Nonlinear Systems. *Int. J. Fuzzy Syst.* **2019**, *21*, 2575–2587. [CrossRef]
39. Zhao, Q.; Duan, G. Concurrent Learning Adaptive Finite-Time Control for Spacecraft with Inertia Parameter Identification under External Disturbance. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3691–3704. [CrossRef]
40. Bu, X.; Luo, R.; Lei, H. Chattering-avoidance discrete-time fuzzy control with finite-time preselected qualities. *IEEE Trans. Fuzzy Syst.* **2024**, *33*, 997–1008. [CrossRef]
41. Bu, X.; Luo, R.; Lei, H. Fuzzy-neural intelligent control with fixed-time pre-configured qualities for electromechanical dynamics of electric vehicles motors. *IEEE Trans. Intell. Transp. Syst.* **2024**, *26*, 1193–1202. [CrossRef]
42. Luo, R.; He, G.; Li, Y.; Bu, X.; Li, Q. Appointed-time Fuzzy Fault-tolerant Control of Hypersonic Flight Vehicles with Flexible Predefined Behaviors. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 13995–14007. [CrossRef]
43. Lin, X.; Shi, X.; Li, S. Adaptive Tracking Control for Spacecraft Formation Flying System via Modified Fast Integral Terminal Sliding Mode Surface. *IEEE Access* **2020**, *8*, 198357–198367. [CrossRef]
44. Wang, Q.; Wang, T.; Dong, C.; Jiang, W. Chained Smooth Switching Control for Morphing Aircraft. *Control Theory Appl.* **2015**, *32*, 949–954.
45. Cheng, H.; Fu, W.; Dong, C.; Wang, Q.; Hou, Y. Asynchronously Finite-Time H∞ Control for Morphing Aircraft. *Trans. Inst. Meas. Control* **2018**, *40*, 4330–4344. [CrossRef]
46. Chen, H.; Wang, P.; Tang, G. Attitude Control Scheme for Morphing Vehicles with Output Error Constraints and Input Saturation. *Acta Aeronaut. Astronaut. Sin.* **2023**, *44*, 408–419.
47. Sheng, H. Research on Strong Robust Control for Flexible Hypersonic Morphing Vehicle. Master's Thesis, National University of Defense Technology, Changsha, China, 2021.
48. Liang, X.; Wang, Q.; Xu, B.; Dong, C. Back-stepping fault-tolerant control for Morphing Unmanned aircraft based on fixed-time observer. *Int. J. Control Autom. Syst.* **2021**, *19*, 3924–3936. [CrossRef]
49. Wu, Z.; Lu, J.; Zhou, Q.; Shi, J. Modified adaptive neural dynamic surface control for Morphing Unmanned aircraft with input and output constraints. *Nonlinear Dyn.* **2017**, *87*, 2367–2383. [CrossRef]
50. Emmanuel, S.A.; Sun, R. Approximation-free prescribed performance control for Nonlinear morphing missile System. *Int. J. Eng. Appl. Sci.* **2016**, *3*, 257584.
51. Wang, Z.; Chang, Y.; Qiu, Y.; Xing, X. RL-based adaptive control for a class of non-affine uncertain stochastic systems with mismatched disturbances. *Commun. Nonlinear Sci. Numer. Simul.* **2024**, *138*, 108191. [CrossRef]
52. Zhang, J.; Wang, S.; Zhou, P.; Zhao, L.; Li, S. Novel Prescribed Performance-Tangent Barrier Lyapunov Function for Neural Adaptive Control of the Chaotic PMSM System by Backstepping. *Int. J. Electr. Power Energy Syst.* **2020**, *121*, 105991. [CrossRef]
53. Wang, F.; Chen, B.; Liu, X.; Lin, C. Finite-time adaptive fuzzy tracking control design for nonlinear systems. *IEEE Trans. Fuzzy Syst.* **2017**, *26*, 1207–1216. [CrossRef]
54. Qian, C.; Lin, W. Non-Lipschitz Continuous Stabilizers for Nonlinear Systems with Uncontrollable Unstable Linearization. *Syst. Control Lett.* **2001**, *42*, 185–200. [CrossRef]
55. Li, M.; Zhang, J.; Li, S.; Wu, F. Adaptive Finite-Time Fault-Tolerant Control for the Full-State-Constrained Robotic Manipulator with Novel Given Performance. *Eng. Appl. Artif. Intell.* **2023**, *125*, 106650. [CrossRef]
56. Wei, Y.; Zhou, P.; Liang, Y.; Wang, Y.; Duan, D. Adaptive finite-time neural backstepping control for multi-input and multi-output state-constrained nonlinear systems using tangent-type nonlinear mapping. *Int. J. Robust Nonlinear Control* **2020**, *30*, 5559–5578. [CrossRef]
57. Wang, F.; Chen, B.; Lin, C.; Zhang, J.; Meng, X. Adaptive Neural Network Finite-Time Output Feedback Control of Quantized Nonlinear Systems. *IEEE Trans. Cybern.* **2018**, *48*, 1839–1848. [CrossRef]
58. Zhang, J.; Wang, S.; Li, S.; Zhou, P. Adaptive Neural Dynamic Surface Control for the Chaotic PMSM System with External Disturbances and Constrained Output. *Recent Adv. Electr. Electron. Eng.* **2020**, *13*, 894–905.

59. Wu, J.; Sun, Y.; Zhao, Q.; Wu, Z.-G. Adaptive neural asymptotic tracking control for a class of stochastic non-strict-feedback switched systems. *J. Frankl. Inst.* **2018**, *359*, 1274–1297. [CrossRef]
60. Chen, K.; Zhu, S.; Wei, C.; Xu, T.; Zhang, X. Output constrained adaptive neural control for generic hypersonic vehi-cles suffering from non-affine aerodynamic characteristics and stochastic disturbances. *Aerosp. Sci. Technol.* **2021**, *111*, 106469. [CrossRef]

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