mathematics

MDPI

Article

Prescribed Fixed-Time Adaptive Neural Control for Manipulators with Uncertain Dynamics and Actuator Failures

Guanyu Lai [1], Sheng Zhou [1], Weijun Yang [2,*], Xiaodong Wang [2,*] and Fang Wang [3]

[1] School of Automation, Guangdong University of Technology, Guangzhou 510006, China; lgy124@foxmail.com (G.L.); 2112104168@mail2.gdut.edu.cn (S.Z.)
[2] School of Mechanical and Electrical Engineering, Guangzhou City Polytechnic, Guangzhou 510405, China
[3] College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China; sandywf75@126.com
* Correspondence: ywj@gcp.edu.cn (W.Y.); wxd@gcp.edu.cn (X.W.)

Abstract: In this paper, a fixed-time adaptive neural control scheme is proposed to solve the prescribed tracking problem of robot manipulators in the presence of uncertain dynamics, and stuck-type actuator failures which are unknown in time, pattern, and values. Technically, the combination of neural networks and adaptive control is used to handle the uncertainties in system dynamics, an adaptive compensation mechanism is designed to accommodate the failures occurring in actuators, and also a systematic design procedure based on the prescribed performance bounds is presented to establish the conditional inequality for ensuring fixed-time stability. With our scheme, it can be proved rigorously that the tracking errors in joint space can always be kept within the prescribed bounds, and converge to a small region of zero in a bounded settling time, in addition to the closed-loop signal boundedness. The proposed scheme is validated through simulations.

Keywords: robust adaptive control; neural networks; robot manipulators; uncertain dynamics; actuator failures

MSC: 37M25

check for updates

Citation: Lai, G.; Zhou, S.; Yang, W.; Wang, X.; Wang, F. Prescribed Fixed-Time Adaptive Neural Control for Manipulators with Uncertain Dynamics and Actuator Failures. *Mathematics* **2023**, *11*, 2925. https://doi.org/10.3390/math11132925

Academic Editor: Ivo Petráš

Received: 18 May 2023
Revised: 26 June 2023
Accepted: 27 June 2023
Published: 29 June 2023

Copyright: © 2023 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

With the growing demand for robots in various domains of modern society, there is an increasing need for improved robot performance, making the research of robot control technology a significant concern. Existing robot control approaches primarily prioritize achieving accurate position control in the presence of actuator faults, but often overlook transient response and settling time. In practical robotic applications, insufficient transient performance can potentially result in collisions between the robot and its environment, thereby compromising system safety.

Poor transient performance can lead to system safety issues in practical applications by causing the robot to crash with the surroundings. To address this issue and achieve desired transient responses while considering various constraints, several controllers have been proposed for robots, as discussed in previous studies [1–3]. In the field of nonlinear control, Barrier Lyapunov Functions (BLFs) have emerged as a novel approach to indirectly enforce state and output constraints. The concept of BLFs was initially introduced in [4,5]. In [4], a BLF-based fuzzy control scheme with an adaptive observer was proposed to address specific challenges in a class of systems. The publication [5] focused on studying the trajectory tracking issue of a particular class of nonlinear dynamic systems that incorporate time-delayed constraints on the entire state.

Uncertain nonlinear elements, such as time-varying parameter values and outside disturbances, frequently affect how robot dynamics systems operate. To tackle these challenges, the utilization of adaptive control techniques in conjunction with neural networks [6–10]

and fuzzy logic approaches [11–13] has been extensively adopted for the purpose of tracking control in uncertain robot systems. In reference [9], a neural-network-based control method for robots is presented, aiming to accomplish accurate tracking of trajectories while ensuring compliance with output constraints and input saturation limitations. Another approach presented in [10] introduces an adaptive admittance method based on an NN controller, ensuring guaranteed trajectory tracking. Additionally, [14] provides a comprehensive overview of advanced neural network control algorithms suitable for nonlinear systems, including robots. In terms of fuzzy control, [13] suggests an adaptive fuzzy control strategy that meets user-specified performance goals as well as fixed-time convergence. It is worth noting that only a limited number of existing control systems address both convergence time and predicted transient performance simultaneously.

In industrial systems, achieving rapid convergence of system states is crucial for enhancing control performance. Earlier investigations have predominantly centered around the convergence time of system [15–17]. As an example, in [15], a fuzzy control strategy is presented, utilizing an adaptive observer to achieve finite-time convergence of a specific category of nonlinear systems characterized by strict feedback. For nonlinear systems with specific matching uncertainties, a flexible finite-time mode control sliding mode strategy is provided in [16]. Moreover, in [18], an appropriate Barrier Lyapunov Function is incorporated into the formulation of a neural network-based control scheme to ensure desired transient performance and prevent deviations in the tracking errors that would violate the output constraints. While the aforementioned research makes substantial contributions to addressing transient behavior and convergence time, they overlook the impact of actuator failure on the overall behavior of the control system. In practical applications, addressing actuator faults has gained considerable attention as a research area to improve the reliability and safety of robots.

Partial loss of effectiveness (PLOE) and total loss of efficacy (TLOE) are the two basic categories into which actuator failures may be divided. In the case of PLOE, the actuator output loses some validity compared to the input. With TLOE, the actuator output becomes unknown regardless of the input. Over the past few decades, several intriguing approaches to fault-tolerant control have been introduced, including multi-model control [19,20], fault detection and diagnostic design [21,22], and sliding mode control [23–26]. Moreover, adaptive control has emerged as a promising technology [27–34], allowing for the dynamic adaptation of controller settings to effectively estimate and compensate for unknown faults and uncertainties in the system. For nonlinear systems, adaptive fault-tolerant controllers based on reverse iteration have been developed in [32–34]. In the context of fault-tolerant robot control, significant progress has been made [35–37]. In [35], a novel controller for cooperative robot systems is proposed, combining multiple individual fault compensators with adaptive control techniques. However, it is important to acknowledge that the complete characterization of failure modes can result in rapid growth in the number of estimated parameters, which can significantly affect the dynamic behavior of the system. To tackle this challenge, a dynamic controller structure is introduced in [36] with the aim of minimizing the potential actuator failure modes, thereby enhancing the effectiveness of parameter adaptation. Nevertheless, developing a fixed-time control approach for a robot with malfunctioning actuators that ensures desired transient performance tracking is still a challenging task.

Motivated by the observation above, in this work, we propose a fixed-time adaptive neural control scheme with prescribed tracking performance for robot manipulators with uncertain dynamics, and unknown actuator failures. The strategy of adaptive control, and its combination with neural networks are used to develop techniques to compensate for actuator failures which are unknown in time, pattern, and values, and deal with uncertain structured or unstructured nonlinear dynamics, and then a prescribed performance bounds (PPB)-based design approach is given to construct the control scheme so that the required fixed-time stability condition can be established. In summary, the work of this study has the following novelties and contributions:

- An actuation scheme is constructed for redundant control inputs so that the possible interaction problem in the control gain matrix can be well-circumvented, and then an adaptive compensation mechanism is raised to accommodate the actuator failures, which are unknown in time, pattern, and values.
- Based on the developed adaptive actuator failure compensation strategy, we further propose a PPB-based adaptive neural control algorithm to establish the conditional inequality of fixed-time stability, so that the response plot of tracking error can be kept within some prescribed bounds, and converges to a residual around zero in a bounded settling time (which can be independent of the initial system states).
- For our raised scheme, an optimization strategy is adopted in the design of adaptive laws for handling the unknown weight matrix of neural networks, based on which it is well achieved that the number of adaptive parameters does not increase with the number of neurons. In this sense, the proposed scheme is computationally attractive.

The rest of this paper is organized as follows. In the second part, the formulation and preliminary work of the problems related to control design are presented. The third part makes a technique for separating the control signal from the fault signal that is based on decoupling. The stability study of the system and the construction of a fixed-time neural network control with transient performance are given. The simulation findings are showcased in the fourth segment. The concluding remarks of the thesis are provided in the fifth section.

2. Problem Statement and Preliminaries

This section describes the control problem to be resolved, and presents some basics on fixed-time stability and neural networks.

2.1. Problem Formulation

The dynamics model of a robot manipulator with redundant actuators can be written as follows:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{1}$$

where $q, \dot{q}, \ddot{q} \in R^n$ are joint variables position, velocity and acceleration vectors; $D(q) \in R^{n\times n}$ is the inertia matrix, $C(q,\dot{q}) \in R^{n\times n}$ is the Coriolis and centrifugal term, $g(q) \in R^{n\times 1}$ is the gravity term, $\tau \in R^{n\times 1}$ is the joint torque vector and n is the degree of freedom.

In this study, we take into account the robot model with unidentified parameters that is provided in the third portion. The design of the controller with fixed-time convergence aims to enable the joint position q in Equation (1) to follow the intended trajectory q_d. The tracking error can simultaneously realize the transitory error specified by the predetermined precision. Figure 1 illustrates the controller design strategy.

2.2. Preliminaries

Property 1 ([38]). *The matrix $D(q)$ in Equation (1) possesses the properties of symmetry and positive definiteness. Additionally, the expression $\dot{D}(q) - 2C(q,\dot{q})$ represents a skew-symmetric matrix, which fulfills the condition $\delta^T(\dot{D}(q) - 2C(q,\dot{q}))\delta = 0$ for any $\delta \in R^n$.*

Assumption 1. *The reference trajectory vector q_d and its first-order derivative $\dot{q}_d$ are both continuous and bounded, where $q_d \in R^n$ and $\dot{q}_d \in R^n$.*

Lemma 1 ([39]). *In the context of a system described by $\dot{x}(t) = g(x)$, where x represents the system state, practical fixed-time stability is achieved when the Lyapunov function $V(x)$ satisfies the following inequality: $\dot{V}(x) \leq -\mu_1 V^{\eta_1}(x) - \mu_2 V^{\eta_2}(x) + \chi$, where, μ_1, μ_2, and χ represent positive constants, while the parameters $\eta_1 \in (0,1)$ and $\eta_2 \in (1,\infty)$. Moreover, the remaining set of the system can be represented as*

$$\Omega_1 = \{x \mid V \leq \min\Big([\chi/(\mu_1(1-\theta))]^{1/\eta_1} \\ [\chi/(\mu_2(1-\theta))]^{1/\eta_2}\Big)\}, \tag{2}$$

given that $\theta \in (0,1)$*, we can derive the expression for the time it takes for the system to converge using the following derivation:*$T \leq T_{\max} = \frac{1}{\mu_1\theta(1-\eta_1)} + \frac{1}{\mu_2\theta(\eta_2-1)}$.

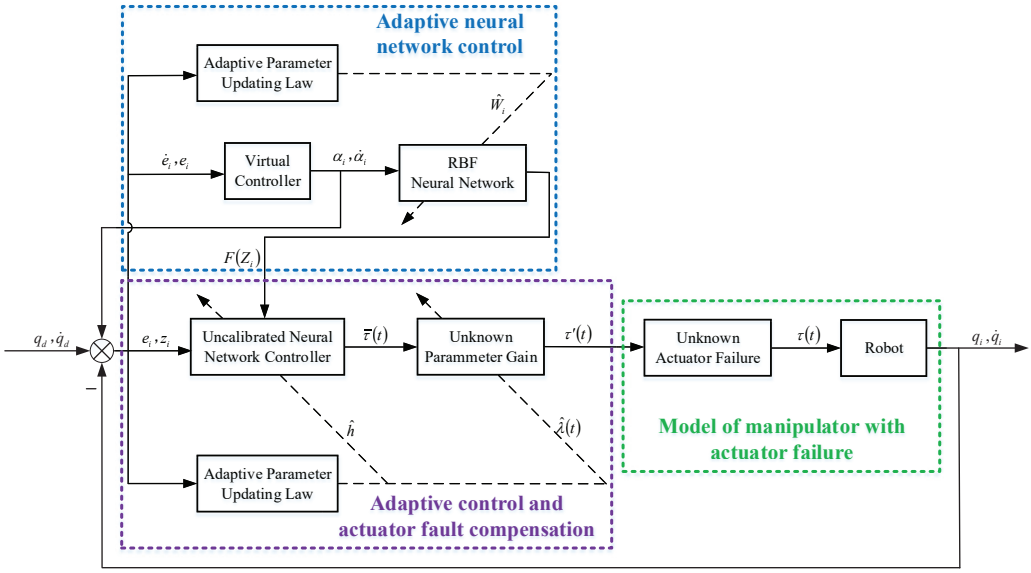


Figure 1. The framework of fixed-time neural network robot controller.

Lemma 2 ([40]). *For every* $x_j \in R$ *and* $d \in R^+$*, the subsequent inequalities hold true:*

$$\sum_{r=1}^{n}|x_j|^d \geq \left(\sum_{j=1}^{n}|x_j|\right)^d, 0 < d \leq 1 \tag{3}$$

$$\sum_{j=1}^{n}|x_j|^d \geq n^{1-d}\left(\sum_{j=1}^{n}|x_j|\right)^d, d > 1. \tag{4}$$

Lemma 3 ([41]). *For any* $Z \in R$ *and* $\epsilon > 0$*, the following inequality holds:*

$$0 \leq |Z| - \frac{Z^2}{\sqrt{Z^2+\epsilon^2}} \leq \epsilon. \tag{5}$$

Remark 1 ([42]). *For any* $\mu \in R$*, a new function is defined in the following manner:*

$$sig^b(\mu) = |\mu|^b sign(\mu), \tag{6}$$

where $b > 0$.

Radial Basis Function Neural Network (RBFNN): In this study, the utilization of RBFNN in the controller design aims to tackle an unspecified system function $f(Z) : R^u \to R$:

$$\hat{f}(Z) = \sum_{i=1}^{N_0}\hat{w}_i s_i(Z) = \hat{W}^T S(Z), \tag{7}$$

where $Z \in \Omega_{Z_0} \subset R^u$, where u represents the size of the NN input vector. The term $\hat{W} \in R^{N_0}$ denotes the estimation of the optimal constant weight vector W^*, with N_0 being the count of chosen NN units. Additionally, $S(Z) = [s_1(Z), s_2(Z), \ldots, s_{N_0}(Z)]^T \in R^{N_0}$

represents the regression vector. Each $s_i(Z)$ is chosen as a typical radial basis function, represented by:

$$s_i(Z, v_i) = \exp\left[\frac{-(Z - v_i)^T(Z - v_i)}{\sigma_i^2}\right], \tag{8}$$

where $v_i \in R^u$ and $\sigma_i \in R$ represent the centroid vector and positive spread value of the ith neural network unit, respectively. By appropriately selecting these parameters, the RBFNN can accurately approximate the continuous function $R^u \to R$ across the bounded region Ω_{Z_0}, assuming a significant number of nodes are employed:

$$f(Z) = W^{*T}S(Z) + \varepsilon(Z), \quad \forall Z \in \Omega_{Z_0} \in R^u, \tag{9}$$

where variable $\varepsilon(Z)$ represents the estimation discrepancy of the neural network. The estimation discrepancy $\varepsilon(Z)$ is limited by a non-negative value $\bar{\varepsilon}$, satisfying $|\varepsilon(Z)| \leq \bar{\varepsilon}$ for all $Z \in \Omega_{Z_0}$. The element that achieves the minimum value of $\varepsilon(Z)$ for every $Z \in \Omega_{Z_0}$ is known as the optimal weight vector:

$$W^* = \arg\min_{W \in R^{N_0}} \left\{ \sup_{Z \in \Omega_{z_0}} \left| f(Z) - \hat{W}^T S(Z) \right| \right\}. \tag{10}$$

3. Control Design

The dynamics of the manipulator with unknown actuator faults are examined in this part, and a novel adaptive approach is suggested to correct for trapped actuator defects. The robustness of the setup is established by utilizing the Lyapunov stability analysis approach before deploying the controller. Algorithm 1 gives the overall simple design idea.

Algorithm 1: New decoupling algorithm: eliminate actuator faults.

Input: Joint position q and joint velocity $\dot{q}$.
Output: Decoupled control input signal $\tau(t)$.

foreach *i=1; i*$\leq m_i$ **do**
  Calculate the torque at the ith joint by Equation (12);
  if *actuator fails* **then**
    set $\sigma_{ij} = 1$;
  else
    set $\sigma_{ij} = 0$;
  end
end
Set fault actuator moment value $\bar{\tau}_{ij}(t)$ and normal actuator moment value $\tau_{ij}^*(t)$;
Obtain the robot joint input by Equation (17);
foreach *j=1; j*$\leq m_i$ **do**
  Introduce and calculate the parameter matrices **B** and **h**;
end
Obtain control signal $\tau(t)$ by Equation (19).
Set parameter λ by Equation (20) and the estimated value $\hat{\lambda}$ of parameter λ.
foreach *i=1; i*$\leq n$ **do**
  Calculate and separate control signals $\hat{\tau}(t)$;
end
Update the control input $\tau(t)$.

3.1. Dynamics of Manipulator with Unknown Actuator Failure

For the ith joint, denoted by $i \in \{1, 2, \ldots, n\}$, in the scenario of concurrent actuation, there exist m_i interconnected actuators. The motion behavior of the ith joint can be characterized in the following manner:

$$D_i(q)\ddot{q} + C_i(q,\dot{q})\dot{q} + G_i(q) = \tau_i, \tag{11}$$

The joint torque is defined as follows in order to reflect the redundancy

$$\tau_i = \tau_{i1} + \tau_{i2} + \cdots + \tau_{im_i}, \tag{12}$$

where τ_{ij} represents the torque applied to the ith joint by the jth actuator. The following describes the situation of an actuator with a TLOE-type failure:

$$\sigma_{ij} = \begin{cases} 1, & \text{if actuator fails;} \\ 0, & \text{otherwise} \end{cases}, \tag{13}$$

where $j = 1, 2, \ldots, m_i$ and σ_{ij} represent the jth actuator associated with the ith joint state, respectively. A common occurrence in actuator malfunction is

$$\bar{\tau}_{ij}(t) = \bar{\tau}_{ij}, t \geq t_{ij}, \tag{14}$$

where $\bar{\tau}_{ij}$ is the constant torque generated by the fault actuator, and t_{ij} is the unknown instantaneous time when the fault occurs.

The underlying principle of the adaptive compensation technique, which tackles uncertain system dynamics and failure characteristics, is as follows.

- Up to $m_i - 1$ actuators can experience instantaneous failure, and even in the absence of knowledge about the specific failure characteristics, the remaining actuators can dynamically adjust their actions to achieve the desired objective.

The torque exerted by the jth actuator on the ith joint of the manipulator can be mathematically represented as follows:

$$\tau_{ij} = (1 - \sigma_{ij})\tau_{ij}^* + \sigma_{ij}\bar{\tau}_{ij}, \tag{15}$$

where the control input is represented by τ_{ij}^*. All actuators must receive an identical control signal or follow an equivalent actuation scheme to form a meaningful actuation design.

$$\tau_{i1}^*(t) = \tau_{i2}^*(t) = \cdots = \tau_{im_i}^*(t) = \tau_i^*(t). \tag{16}$$

By substituting Equation (15) into the joint input of the robot, we acquire

$$\tau_i = \sum_{j=1}^{m_i}(1 - \sigma_{ij})\tau_i^* + \sum_{j=1}^{m_i}\sigma_{ij}\bar{\tau}_{ij}. \tag{17}$$

We can express the matrix $\mathbf{B}$ and the vector $\mathbf{h}$ as follows:

$$\begin{aligned} \mathbf{B} &= \operatorname{diag}(d_1, d_2, \ldots, d_n), d_i = \sum_{j=1}^{m_i}(1 - \sigma_{ij}) \\ \mathbf{h}(t) &= (h_1, h_2, \ldots, h_n)^T, h_i = \sum_{j=1}^{m_i}\sigma_{ij}\bar{\tau}_{ij}. \end{aligned} \tag{18}$$

Based on Equations (17) and (18), we can represent the control signal $\boldsymbol{\tau}(t)$ as follows:

$$\boldsymbol{\tau}(t) = \mathbf{B}(t)\tau^*(t) + \mathbf{h}(t), \tag{19}$$

where $\mathbf{B}$ is a positive definite matrix of size $n \times n$. The desired joint input signal to be designed is denoted as τ^*, which is an $n \times 1$ vector. Additionally, $\mathbf{h}(t)$ represents the constant torque contributed by the failed actuator, forming a constant $n \times 1$ vector.

An unidentified parameter vector was used in the control input design to make it easier. We introduce the parameter λ, which is defined by the subsequent equation:

$$\begin{aligned} \lambda_i &= \frac{1}{d_i} \\ \tau^*(t) &= \hat{\lambda}\hat{\tau}(t), \end{aligned} \tag{20}$$

where $\hat{\lambda}$ denotes the estimation of λ, and $\widetilde{\lambda_i}$ refers to the difference between the estimated value $\hat{\lambda}_i$ and the actual value λ_i. Mathematically, it can be expressed as $\widetilde{\lambda_i} = \hat{\lambda}_i - \lambda_i$. Additionally, the control input $\hat{\tau}(t)$ will be used in place of the designed control input $\tau^*(t)$. By incorporating Equations (18) and (20) into Equation (19), we can accomplish the task of isolating the control signal $\hat{\tau}(t)$, as demonstrated in the following procedure:

$$\begin{aligned} \boldsymbol{\tau}(t) &= \mathbf{B}(t)\tau^*(t) + \mathbf{h}(t) \\ &= \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} \tau_1^* \\ \tau_2^* \\ \vdots \\ \tau_n^* \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\lambda_1} & & & \\ & \frac{1}{\lambda_2} & & \\ & & \ddots & \\ & & & \frac{1}{\lambda_n} \end{bmatrix} \begin{bmatrix} \left(\tilde{\lambda}_1 + \lambda_1\right)\hat{\tau}_1 \\ \left(\tilde{\lambda}_2 + \lambda_2\right)\hat{\tau}_2 \\ \vdots \\ \left(\tilde{\lambda}_n + \lambda_n\right)\hat{\tau}_n \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} \\ &= \begin{bmatrix} \hat{\tau}_1 \\ \hat{\tau}_2 \\ \vdots \\ \hat{\tau}_n \end{bmatrix} + \begin{bmatrix} d_1\tilde{\lambda}_1\hat{\tau}_1 \\ d_2\tilde{\lambda}_2\hat{\tau}_n \\ \vdots \\ d_n\tilde{\lambda}_n\hat{\tau}_n \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix}. \end{aligned}$$

Taking this into account, the control input arising from a defective actuator can be redefined as follows:

$$\boldsymbol{\tau}(t) = \hat{\tau}(t) + \mathbf{h}(t) + \mathbf{r}(t), \tag{21}$$

where

$$\mathbf{r}(t) = \begin{pmatrix} d_1\tilde{\lambda}_1\hat{\tau}_1 \\ d_2\tilde{\lambda}_2\hat{\tau}_2 \\ \vdots \\ d_n\tilde{\lambda}_n\hat{\tau}_n \end{pmatrix}. \tag{22}$$

By incorporating Equation (21) into Equation (1), we can express the manipulator dynamics:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \hat{\tau}(t) + \mathbf{h}(t) + \mathbf{r}(t). \tag{23}$$

Remark 2. *Following Equation (23), the control signals for the robotic system under an unforeseen actuator malfunction are denoted by $\hat{\tau}(t)$, $\mathbf{h}(t)$, and $\mathbf{r}(t)$. The parameter $\mathbf{B}(t)$ is subject to uncertainty and does not have a predefined range, while the quantity $\mathbf{h}(t)$ is both limited and of an unspecified nature.*

3.2. Fixed-Time Neural Network Controller and Stability Analysis

Prior to proceeding with the controller design, provided below is a description of the tracking error signals:

$$e = q - q_d \quad z = \dot{q} - \alpha, \tag{24}$$

where $e = [e_1, e_2, \ldots, e_n]^T \in R^n$ and $z = [z_1, z_2, \ldots, z_n]^T \in R^n$ represent the position tracking error and velocity tracking error vectors of the robot manipulator, respectively. $q(t)$ and $q_d(t) \in R^n$ represent the actual and desired trajectories, respectively. Furthermore, $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T \in R^n$ represents the virtual controllers, which will be determined later.

To limit the position deviation e_i in various orientations, the boundary functions ξ_{1i} and ξ_{2i} are devised as follows:

$$\begin{array}{c} \xi_{1i} = -\left[(\varrho_{0,1i} - \varrho_{\infty,1i})e^{-k_{1i}t} + \varrho_{\infty,1i}\right] \\ \xi_{2i} = (\varrho_{0,2i} - \varrho_{\infty,2i})e^{-k_{2i}t} + \varrho_{\infty,2i} \end{array}, \tag{25}$$

where $\varrho_{0,ji}, \varrho_{\infty,ji}, k_{ji} (j = 1, 2, i = 1, 2, \ldots, n)$ are positive constants that are chosen appropriately.

Remark 3. *The functions ξ_{2i} and ξ_{1i} define the upper and lower limits of the position deviation e_i, respectively, with $e_i = q_i - q_{di}$. This means that the tracking error e_i is restricted to the range $\xi_{1i} < e_i < \xi_{2i}$. By appropriately selecting the variables $\varrho_{0,ji}$, $\varrho_{\infty,ji}$, k_{ji}, and utilizing the monotonic nature of the exponential function, we can evaluate the tracking performance at a steady state based on Equation (25).*

The tracking error vector e undergoes a coordinate transformation as follows:

$$\vartheta_a = \left[\frac{e_1}{\xi_{11}}, \frac{e_2}{\xi_{12}}, \ldots, \frac{e_n}{\xi_{1n}}\right]^T, \vartheta_b = \left[\frac{e_1}{\xi_{21}}, \frac{e_2}{\xi_{22}}, \ldots, \frac{e_n}{\xi_{2n}}\right]^T. \tag{26}$$

Then, we define a new vector ϑ as $\vartheta = [\vartheta_1, \vartheta_2, \ldots, \vartheta_n]^T$, where each element ϑ_i corresponds to

$$\vartheta_i = H_i(e_i)\vartheta_{bi} + (1 - H_i(e_i))\vartheta_{ai}, \quad i = 1, 2, \ldots, n \tag{27}$$

The elements ϑ_{ai} and ϑ_{bi} in Equation (27) represent the ith components of the vectors ϑ_a and ϑ_b, respectively. The function $H_i(e_i)$ can be expressed by the following definition:

$$H_i(e_i) = \begin{cases} 1 & e_i \geq 0 \\ 0 & e_i < 0 \end{cases}. \tag{28}$$

Then, we consider a Lypunov candidate

$$V_1 = \sum_{i=1}^{n}\left[\frac{H_i}{2}\vartheta_{b_i}^2 + \frac{(1 - H_i)}{2}\vartheta_{a_i}^2\right]. \tag{29}$$

By combining Equations (26) and (28), and their time derivatives, we can obtain the following expression:

$$\dot{V}_1 = \sum_{i=1}^{n}\left[H_i\vartheta_{b_i}\dot{\vartheta}_{b_i} + (1 - H_i)\vartheta_{a_i}\dot{\vartheta}_{a_i}\right]. \tag{30}$$

From (24), we have

$$\dot{e}_i = \dot{q}_i - \dot{q}_{di} = z_i + \alpha_i - \dot{q}_{di}. \tag{31}$$

Substituting (26) and (31) into (30), we have

$$\dot{V}_1 = \sum_{i=1}^{n}\left[\frac{H_i\vartheta_{b_i}}{\xi_{2i}}\left(z_i + \alpha_i - \dot{q}_{d_i} - e_i\frac{\dot{\xi}_{2i}}{\xi_{2i}}\right)\right] + \sum_{i=1}^{n}\left[\frac{(1 - H_i)\vartheta_{a_i}}{\xi_{1i}}\left(z_i + \alpha_i - \dot{q}_{d_i} - e_i\frac{\dot{\xi}_{1i}}{\xi_{1i}}\right)\right]. \tag{32}$$

The formulation of the virtual controller α_i is outlined below:

$$\begin{aligned}\alpha_i =& \dot{q}_{d_i} - k_{1i}e_i - H_i k_{2i}\vartheta_{b_i}^2 e_i - (1-H_i)k_{2i}\vartheta_{a_i}^2 e_i - H_i k_{3i}\left(\frac{1}{2}e_i^2 + \frac{1}{2}\xi_{2i}^2\right)^{\frac{1}{2}} - \\ & \delta_i e_i + (1-H_i)k_{3i}\left(\frac{1}{2}e_i^2 + \frac{1}{2}\xi_{1i}^2\right)^{\frac{1}{2}},\end{aligned} \tag{33}$$

where

$$\delta_i = \sqrt{\left(\frac{\dot{\xi}_{1i}}{\xi_{1i}}\right)^2 + \left(\frac{\dot{\xi}_{2i}}{\xi_{2i}}\right)^2 + K_{a_i}}, \tag{34}$$

and $k_{1i}, k_{2i}, k_{3i}, k_{ai}$ are all proper positive constants.

By replacing (33) with Equation (32), we obtain the resulting expression:

$$\begin{aligned}\dot{V}_1 =& \sum_{i=1}^{n}\left[\left(\frac{H_i e_i}{\xi_{2i}^2} + \frac{(1-H_i)e_i}{\xi_{1i}^2}\right)z_i\right] - \sum_{i=1}^{n}\left[\frac{k_{1i}H_i e_i^2}{\xi_{2i}^2} + \frac{k_{1i}(1-H_i)e_i^2}{\xi_{1i}^2}\right] - \\ & \sum_{i=1}^{n}\left[k_{2i}H_i^2\vartheta_{b_i}^4 + (1-H_i)^2 k_{2i}\vartheta_{a_i}^4\right] + \sum_{i=1}^{n}\left[\frac{k_{3i}H_i^2\vartheta_{bi}\sqrt{\frac{1}{2}e_i^2 + \frac{1}{2}\xi_{2i}^2}}{\xi_{2i}}\right] - \\ & \sum_{i=1}^{n}\left[\frac{k_{3i}(1-H_i)^2\vartheta_{ai}\sqrt{\frac{1}{2}e_i^2 + \frac{1}{2}\xi_{1i}^2}}{\xi_{1i}}\right] - \sum_{i=1}^{n}\left[H_i\vartheta_{b_i}^2\left(\delta_i + \frac{\dot{\xi}_{2i}}{\xi_{2i}}\right) + \frac{(1-H_i)\vartheta_{ai}^2}{1-\vartheta_{a_i}^2}\left(\delta_i + \frac{\dot{\xi}_{1i}}{\xi_{1i}}\right)\right].\end{aligned} \tag{35}$$

When $e_i \geq 0$, we can derive from Equations (25) and (26):

$$-\sum_{i=1}^{n}\frac{k_{3i}H_i^2\vartheta_{bi}\sqrt{\frac{1}{2}e_i^2 + \frac{1}{2}\xi_{2i}^2}}{\xi_{2_i}} \leqslant -\sum_{i=1}^{n}K_{3i}H_i^2\left(\vartheta_{b_i}^2\right)^{\frac{3}{4}}. \tag{36}$$

When $e_i < 0$, we can obtain from Equations (25) and (26):

$$\sum_{i=1}^{n}\frac{k_{3i}(1-H_i)^2\vartheta_{a_i}\sqrt{\frac{1}{2}e_i^2 + \frac{1}{2}\xi_{ij}^2}}{\xi_{1i}} \leqslant -\sum_{i=1}^{n}k_{3i}(1-H_i)^2\left(\vartheta_{a_i}^2\right)^{\frac{3}{4}}. \tag{37}$$

Using Equation (33), we can derive the subsequent inequalities as follows:

$$\delta_i + \frac{\dot{\xi}_{2i}}{\xi_{2i}} \geqslant 0 \quad \delta_i + \frac{\dot{\xi}_{1i}}{\xi_{1i}} \geqslant 0. \tag{38}$$

Substituting (36)–(38) into (35), we can find that:

$$\dot{V}_1 \leq \sum_{i=1}^{n}\left[-k_{1i}\vartheta_i^2 - k_{2i}\left(\vartheta_i^2\right)^2 - k_{3i}\left(\vartheta_i^2\right)^{\frac{3}{4}} + \zeta_i e_i z_i\right] \tag{39}$$

where

$$\zeta_i = \frac{H_i}{\xi_{2i}^2} + \frac{1-H_i}{\xi_{1i}^2}. \tag{40}$$

Subsequently, a potential Lyapunov function can be formulated in the following manner:

$$V_2 = V_1 + \frac{1}{2}z^T D z + \sum_{i=1}^{n}\frac{d_i}{2c_i}\tilde{\lambda}_i^2 + \sum_{i=1}^{n}\frac{1}{2}\tilde{h}_i^2, \tag{41}$$

where $\tilde{h}$ represents the estimation discrepancy of h for the estimated $\hat{h}$ and actual h.

Consider the derivative of (41), along with (23), (24), and Property 1 to produce

$$\dot{V}_2 = \dot{V}_1 + z^T[\hat{\tau} + h + r + F(z)] + \sum_{i=1}^{n}\frac{d_i}{c_i}\tilde{\lambda}_i\dot{\tilde{\lambda}}_i + \sum_{i=1}^{n}\tilde{h}_i\dot{\tilde{h}}_i. \tag{42}$$

The vector that will be approximated by the RBFNNs is given by the following definition:

$$F(Z) = -D\dot{\alpha} - C\alpha - G, \tag{43}$$

where the symbols D, C, and G are abbreviations for the terms $D(q)$, $C(q,\dot{q})$, and $G(q)$, respectively. The input vector, denoted as Z, can be defined as $Z = \left[q^T, \dot{q}^T, \alpha^T, \dot{\alpha}^T\right]^T \in R^{4n}$. Moreover, the vector $F(Z)$ represents the concatenation of individual functions $f_i(Z)$, and is defined as $F(Z) = [f_1(Z), f_2(Z), \ldots, f_n(Z)]^T$.

By utilizing Equation (41), we can deduce the subsequent equalities:

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + z^T[\hat{\tau} + F(z)] + \sum_{i=1}^{n} z_i r_i + \sum_{i=1}^{n} z_i h_i + \sum_{i=1}^{n} \frac{d_i}{c_i}\tilde{\lambda}_i\dot{\tilde{\lambda}}_i + \sum_{i=1}^{n}\tilde{h}_i\dot{\tilde{h}}_i \\ &= \dot{V}_1 + z^T[\hat{\tau} + F(z)] + \sum_{i=1}^{n}\tilde{\lambda}_i\frac{d_i}{c_i}\left(\dot{\tilde{\lambda}}_i + z_i c_i \hat{\tau}_i\right) + \sum_{i=1}^{n} h_i\left(-\dot{\hat{h}}_i + z_i\right) + \hat{h}_i\dot{\tilde{h}}_i.\end{aligned} \tag{44}$$

For Equation (44), actuator fault compensation, adaptive rules with unknown parameters $\hat{\lambda}$ and $\hat{h}$ can be designed.

$$\dot{\tilde{\lambda}}_i = -z_i c_i \hat{\tau}_i \tag{45}$$

$$\dot{\tilde{h}}_i = z_i. \tag{46}$$

By utilizing Equations (42)–(44), we can deduce the following relationships:

$$\dot{V}_2 = \dot{V}_1 + z^T[\hat{\tau} + F(z)] + \hat{h}_i\dot{\tilde{h}}_i. \tag{47}$$

Then, we can employ n radial basis function neural networks (RBFNNs) to approximate the terms $f_i(Z)$, where $i = 1, 2, \ldots, n$, in the following manner:

$$\begin{aligned}\hat{f}_i(Z) &= \hat{W}_i^T S_i(Z) \\ f_i(Z) &= W_i^{*T} S_i(Z) + \varepsilon_i.\end{aligned} \tag{48}$$

In this context, $W_i^{*T} \in R^{N_i}$ and $\hat{W}_i^T \in R^{N_i}$ indicate the optimal weight vector and the estimated weight vector of the neural network, respectively. N_i denotes the number of nodes in the ith neural network. The vector $S_i(Z) \in R^{N_i}$ symbolizes the fundamental function vector. The discrepancy $\varepsilon_i \in R$ corresponds to the estimation error of the ith neural network and is constrained by $\varepsilon_i < \bar{\varepsilon}_i$.

The neural network controller can be designed in the following manner:

$$\hat{\tau}_i = -k_{4i}z_i - \hat{h}_i - \zeta_i e_i - \frac{\hat{f}_i^2 z_i}{\sqrt{\hat{f}_i^2 z_i^2 + \varepsilon_{1i}^2}} - k_{5i}z_i^3 - k_{6i}\,\mathrm{sig}^{0.5}(z_i), \tag{49}$$

where $k_{4i} > \frac{1}{2}$, $k_{5i} > \frac{1}{2}$ and $k_{5i} > 0$. The term $\mathrm{sig}^{0.5}(z_i)$ carries the significance as described in Remark 1.

The formulation of the adaptive law for updating the neural network weights can be described as

$$\dot{\hat{W}}_i = Q_i(z_i S_i(Z) - m_i\hat{W}_i), \tag{50}$$

the matrix Q_i is a positive definite diagonal matrix, while the parameter m_i represents a positive constant.

Remark 4. *When analyzing the unknown robot arm described in Equation (23), the tracking error e_i defined in Equation (24) can achieve convergence within a fixed time interval if the error remains within the specified upper and lower bounds as stated in Equation (25). This desirable behavior can*

be achieved by utilizing the NN controller stated in (49) together with the adaptive law presented in (50).

Theorem 1. *When the controller given by Equation (49) and the adaptive law described in Equation (50) are employed for the unknown manipulator defined in Equation (23), it ensures that the tracking error stated in Equation (24) remains within the specified bounds, namely $\xi_{1i} < e_i < \xi_{2i}$. Additionally, it can be inferred that all signals within the closed-loop system are bounded.*

Proof of Theorem 1. The Lyapunov function is constructed in the following manner:

$$V = V_2 + \frac{1}{2}\sum_{i=1}^{n} \tilde{W}_i^T Q_i^{-1} \tilde{W}_i, \tag{51}$$

the term $\tilde{W}_i$, which is given by the difference between W_i^* and $\hat{W}_i$, represents the estimation error of the optimal neural network weight W_i^*.

By differentiating V and combining it with Equations (47) and (50), we can derive the following expression:

$$\dot{V} = \dot{V}_1 + z^T[\hat{\tau} + F(Z)] - \sum_{i=1}^{n}\left[m_i \tilde{W}_i^T \dot{\hat{W}}_i - \hat{h}_i \dot{\hat{h}}_i\right]. \tag{52}$$

By substituting Equations (39), (48) and (49) into Equation (52), we can derive the following expression:

$$\begin{aligned}\dot{V} \leq & \sum_{i=1}^{n}\left[-k_{1i}\vartheta_i^2 - k_{2i}\left(\vartheta_i^2\right)^2 - K_{3i}\left(\vartheta_i^2\right)^{\frac{3}{4}}\right] + \sum_{i=1}^{n}\left[-k_{4i}z_i^2 - -K_{5i}z_i^4 - k_{6i}\left(z_i^2\right)^{\frac{3}{4}}\right] + \\ & \sum_{i=1}^{n}\left[z_i\varepsilon_i + m_i\tilde{W}_i^T\hat{W}_i\right] + \sum_{i=1}^{n}\left[\hat{f}_i z_i - \frac{\hat{f}_i^2 z_i^2}{\sqrt{\hat{f}_i^2 z_i^2 + \epsilon_i^2}}\right].\end{aligned} \tag{53}$$

Using Lemma 3, we can reach the following conclusion:

$$\hat{f}_i z_i - \frac{\hat{f}_i^2 z_i^2}{\sqrt{\hat{f}_i^2 z_i^2 + \epsilon_i^2}} \leqslant \left|\hat{f}_i z_i\right| - \frac{\hat{f}_i^2 z_i^2}{\sqrt{\hat{f}_i^2 z_i^2 + \epsilon_i^2}} \leqslant \epsilon_i. \tag{54}$$

The following results can be obtained for the terms $z_i\varepsilon_i$ and $m_i\tilde{W}_i^T\hat{W}_i$:

$$z_i\varepsilon_i \leq |z_i||\varepsilon_i| \leq \frac{1}{2}\left(z_i^2 + \varepsilon_i^2\right) \leq \frac{1}{2}z_i^2 + \frac{1}{2}\bar{\varepsilon}_i^2 \tag{55}$$

$$\begin{aligned} m_i\tilde{W}_i^T\hat{W}_i &= m_i\tilde{W}_i^T\left(W_i^* - \tilde{W}_i\right) = -m_i\left\|\tilde{W}_i\right\|^2 + m_i\tilde{W}_i^T W_i^* \\ &\leq m_i\left(\frac{\left\|W_i^*\right\|^2}{2} - \frac{\left\|\tilde{W}_i\right\|^2}{2}\right).\end{aligned} \tag{56}$$

By substituting Equations (55)–(57) into Equation (53), the following expression is obtained:

$$\begin{aligned}\dot{V} \leq & \sum_{i=1}^{n}\left[-k_{1i}\vartheta_i^2 - k_{2i}\left(\vartheta_i^2\right)^2 - K_{3i}\left(\vartheta_i^2\right)^{\frac{3}{4}}\right] + \\ & \sum_{i=1}^{n}\left[-k_{4i}z_i^2 - K_{5i}z_i^4 - k_{6i}\left(z_i^2\right)^{\frac{3}{4}}\right] + \\ & \sum_{i=1}^{n}\left[m_i\left(\frac{\left\|W_i^*\right\|^2}{2} - \frac{\left\|\tilde{W}_i\right\|^2}{2}\right)\right] + \\ & \sum_{i=1}^{n}\left[\frac{1}{2}z_i^2 + \frac{1}{2}\bar{\varepsilon}_i^2 + \epsilon_i\right].\end{aligned} \tag{57}$$

Based on Equation (57), we can obtain the following inequality by deduction:

$$\begin{aligned}\dot{V} \leq & \sum_{i=1}^{n}(-2k_{1i}) - \min\left(\frac{2k_{4i}-1}{\lambda_{\max}(D)}\right)\frac{1}{2}z^T D z - \min\left(\frac{m_i}{\lambda_{\max}\left(Q_i^{-1}\right)}\right)\sum_{i=1}^{n}\left(\frac{1}{2}\tilde{W}_i^T Q_i^{-1}\tilde{W}_i\right) \\ & + \Lambda_1,\end{aligned} \tag{58}$$

where $\Lambda_1 = \sum_{i=1}^{n}\left[\frac{1}{2}\bar{\varepsilon}_i^2 + \frac{m_i\left\|W_i^2\right\|^2}{2} + \epsilon_i\right]$ can be considered a positive constant with an upper bound, where $\lambda_{\max}(\cdot)$ represents the maximum eigenvalue of the term $(\cdot)$.

Therefore, the desired result can be obtained as follows:

$$\dot{V} \leq -\psi_1 V + \Lambda_1, \tag{59}$$

where $\psi_1 = \min\left(k_{1i}, \frac{2k_{4i}-1}{\lambda_{\max}(D)}, \frac{m_i}{\lambda_{\max}\left(Q_i^{-1}\right)}\right)$.

By multiplying both sides of Equation (59) by $e^{\psi_1 t}$ and integrating both sides over the interval $[0, t]$, we obtain the following equation:

$$V \leq \left[V(0) - \frac{\Lambda_1}{\psi_1}\right]e^{-\psi_1 t} + \frac{\Lambda_1}{\psi_1} \leq V(0) + \frac{\Lambda_1}{\psi_1}. \tag{60}$$

The bounded nature of the Lyapunov function $V(t)$ can be inferred from the aforementioned inequality. By combining the definitions of $V(t)$ in (29), (53) and (51), we can conclude the boundedness of ϑ_i^2, z_i, and $\tilde{W}_i$, respectively. Thus, it can be guaranteed that $|\vartheta_i| < 1$. As a result, the tracking constraints specified by $\xi_{1i} < e_i < \xi_{2i}$ will not be violated, as indicated by Equation (26). Considering Equation (10) and Assumption 1, we can deduce that the variable q remains bounded. Furthermore, taking into account the boundedness of ϑ_i, ξ_{1i}, ξ_{2i}, $\dot{\xi}_{1i}$, and $\dot{\xi}_{2i}$, we can conclude that the virtual controller α_i is bounded as well. Additionally, considering the finite nature of $\hat{W}_i$ and the vector of Gaussian basis functions $S_i(Z)$, it can be demonstrated that the controller described in Equation (49) remains bounded. Thus, it can be inferred that all signals within the closed-loop system are limited in amplitude. □

Corollary 1. *Through further investigation, we can not only achieve fast convergence within a predetermined time, but also accurately restrict the tracking deviation e_i. Furthermore, we can calculate the exact time needed for the error to converge to a negligible value near zero.*

Proof of Corollary 1. The following inequality may be derived from (57) in a manner similar to the derivation of (58):

$$\begin{aligned}\dot{V} \leq \sum_{i=1}^{n}\left[-k_{2i}\left(\vartheta_i^2\right)^2 - K_{3i}\left(\vartheta_i^2\right)^{\frac{3}{4}} - K_{5i}z_i^4 - k_{6i}\left(z_i^2\right)^{\frac{3}{4}}\right] + \\ \sum_{i=1}^{n}\left[\frac{1}{2}z_i^2 + \frac{1}{2}\bar{\varepsilon}_i^2 + \frac{m_i\left\|W_i^*\right\|^2}{2} + \epsilon_i\right]\end{aligned}. \tag{61}$$

For the term $\frac{1}{2}z_i^2$, we have:

$$\frac{1}{2}z_i^2 \leq \frac{1}{2}z_i^4 + \frac{1}{8}. \tag{62}$$

Hence, it can be concluded that the norm of $\tilde{W}_i$ is limited by $\|\tilde{W}_i\| \leq \bar{\tilde{W}}_i$, where $\bar{\tilde{W}}_i$ represents an unspecified positive constant. By replacing Equation (62) with Equation (61), we derive:

$$\begin{aligned}\dot{V} \leq \sum_{i=1}^{n}\left[-k_{2i}\left(\vartheta_i^2\right)^2 - K_{3i}\left(\vartheta_i^2\right)^{\frac{3}{4}} - \left(k_{5i} - \frac{1}{2}\right)z_i^4 - k_{6i}\left(z_i^2\right)^{\frac{3}{4}}\right] - \\ \sum_{i=1}^{n}\left[k_{7i}\left(\left\|\tilde{W}_i\right\|^2\right)^2 + k_{8i}\left(\left\|\tilde{W}_i\right\|^2\right)^{\frac{3}{4}}\right] + \Lambda_2\end{aligned}, \tag{63}$$

where $\Lambda_2 = \sum_{i=1}^{n}\left[K_{7i}\bar{\tilde{W}}_i^4 + K_{8i}\bar{\tilde{W}}_i^{\frac{3}{2}} + \frac{1}{8} + \frac{1}{2}\bar{\varepsilon}_i^2 + \epsilon_i + \frac{m_i\left\|W_i^*\right\|^2}{2}\right]$.

Remark 5. *To conduct the fixed-time stability analysis, we made the assumption that the norm of $\tilde{W}_i$ is limited by $\|\tilde{W}_i\| \leq \bar{\tilde{W}}_i$. However, the integration of the term $k_{7i}(\|\tilde{W}_i\|^2)^2 + k_{8i}(\|\tilde{W}_i\|^2)^{\frac{3}{4}}$ to the positive constant Λ_2 lacks a theoretical basis. In our work, we establish the inequality for the Lyapunov function similar to (46). Therefore, we can ascertain the finite bound of $\|\tilde{W}_i\|$ rather than making assumptions.*

By utilizing Equation (63) and applying Lemma 2, we can deduce the following conclusion:

$$\dot{V} \leq -\psi_2 V^2 - \psi_3 V^{\frac{3}{4}} + \Lambda_2, \tag{64}$$

where $\psi_2 = \frac{1}{3n}\min\left[4k_{2i}, \frac{4k_{5i}-2}{\lambda_{\max}^2(D)}, \frac{4k_{7i}}{\lambda_{\max}^2\left(Q_i^{-1}\right)}\right]$, $\psi_3 = \min\left[\frac{k_{3i}}{\left(\frac{1}{2}\right)^{\frac{3}{4}}}, \frac{k_{6i}}{\left(\frac{1}{2}\right)^{\frac{3}{4}}\lambda_{\max}^{\frac{3}{4}}(D)}, \frac{k_{8i}}{\left(\frac{1}{2}\right)^{\frac{3}{4}}\lambda_{\max}^{\frac{3}{4}}\left(Q_i^{-1}\right)}\right]$.

The inequality form derived in Lemma 1 has been established thus far, demonstrating that the dynamical system achieves fixed-time convergence irrespective of the initial conditions. Furthermore, during the specified time interval, the Lyapunov function V of the system will approach a bounded region defined as follows:

$$\Omega_g = \left\{x \mid V \leq \min\left[\left(\frac{\Lambda_2}{\psi_3(1-\theta_1)}\right)^{\frac{4}{3}}, \left(\frac{\Lambda_2}{\psi_2(1-\theta_1)}\right)^{\frac{1}{2}}\right]\right\}. \tag{65}$$

The time required for convergence can be expressed as follows:

$$T_{g\max} = \frac{4}{\psi_3\theta_1} + \frac{1}{\psi_2\theta_1}, \tag{66}$$

where θ_1 is a positive constant within the range $(0, 1)$, we can attain both rapid convergence within a predetermined time and precise restriction of the tracking deviation e_i. The constraint is given by $\xi_{1i} < e_i < \xi_{2i}$, which is a more precise specification compared to the compact set mentioned in Equation (65). It is worth noting that larger values of ψ_1 and ψ_2 also contribute to a reduction in the convergence time. □

4. Simulation Study

To assess the effectiveness of the suggested controller, a demonstration involving a robotic system with two degrees of freedom will be employed.

4.1. Robot Kinematics

The system block diagram of the mechanical arm is shown in Figure 2. In the figure, we can see that each joint of the robotic arm is driven by m_i drives. The torque output of each joint of the manipulator is generated by m_i actuators.

However, due to certain constraints and limitations, we only use two actuators per joint when simulating the system. The use of multiple actuators in a robotic system introduces additional complexity to the controller design. It requires the development of appropriate control algorithms to manage and adjust the output of each actuator. This may involve multi-variable control, coordination control and optimization problems, which greatly increases the difficulty of algorithm design and debugging. During the experiment, achieving coordination and synchronization between multiple actuators was also a challenging task. When designing the control algorithm, it is necessary to consider the interaction and mutual influence between each actuator to ensure the coordination and synchronization of the motion of each actuator, so as to achieve the ideal motion trajectory and stability.

Therefore, in our simulation, we chose to use a redundant robotic arm with two degrees of freedom, where each joint is driven by two actuators. In robotic systems, redundancy means having more than the minimum number of actuators needed to complete a particular task. In our example, there are two actuators for each joint, although only one actuator is actually needed to drive that joint. Therefore, we call this robotic arm system a redundant robotic arm.

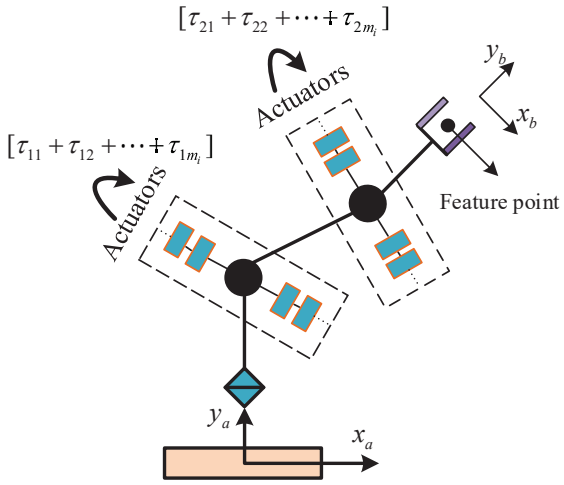


Figure 2. Schematic diagram of redundant robotic arm system.

For the redundant manipulator with two degrees of freedom, each joint is driven by two actuators joint, the joint Angle q_{1_1}, q_{1_2}, q_{2_1} and q_{2_2}. Where q_{1_1}, q_{1_2} are the first joint point of view, q_{2_1} and q_{2_2} represent the second joint point of view.

According to the geometric structure of the manipulator and the length of the connecting rod, the forward kinematics equation of the end-effector position is derived. Assuming that the connecting rod lengths of the mechanical arm are L_1 and L_2, respectively, the coordinates of the end effector position can be represented as

$$\begin{aligned} x &= L_1 \cos(q_{1_1}) + L_2 \cos(q_{1_1} + q_{2_1}) + L_2 \cos(q_{1_1} + q_{2_1} + q_{2_2}) \\ y &= L_1 \sin(q_{1_1}) + L_2 \sin(q_{1_1} + q_{2_1}) + L_2 \sin(q_{1_1} + q_{2_1} + q_{2_2}) \\ z &= \text{specific end-effector height or offset} \end{aligned} . \tag{67}$$

The partial derivative of position with respect to the joint angle is obtained by derivation of the forward kinematics equation. According to the chain rule, we calculate the coordinates of the end-effector position:

$$\begin{aligned}
\frac{\partial x}{\partial q_{1_1}} &= -L_1 \sin(q_{1_1}) - L_2 \sin(q_{1_1} + q_{2_1}) - L_2 \sin(q_{1_1} + q_{2_1} + q_{2_2}) \\
\frac{\partial x}{\partial q_{1_2}} &= -L_2 \sin(q_{1_1} + q_{2_1} + q_{2_2}) \\
\frac{\partial x}{\partial q_{2_1}} &= -L_2 \sin(q_{1_1} + q_{2_1}) - L_2 \sin(q_{1_1} + q_{2_1} + q_{2_2}) \\
\frac{\partial x}{\partial q_{2_2}} &= -L_2 \sin(q_{1_1} + q_{2_1} + q_{2_2}) \\
\frac{\partial y}{\partial q_{1_1}} &= L_1 \cos(q_{1_1}) + L_2 \cos(q_{1_1} + q_{2_1}) + L_2 \cos(q_{1_1} + q_{2_1} + q_{2_2}) \\
\frac{\partial y}{\partial q_{1_2}} &= L_2 \cos(q_{1_1} + q_{2_1} + q_{2_2}) \\
\frac{\partial y}{\partial q_{2_1}} &= L_2 \cos(q_{1_1} + q_{2_1}) + L_2 \cos(q_{1_1} + q_{2_1} + q_{2_2}) \\
\frac{\partial y}{\partial q_{2_2}} &= L_2 \cos(q_{1_1} + q_{2_1} + q_{2_2}) \\
\frac{\partial z}{\partial q_{1_1}} &= 0 \\
\frac{\partial z}{\partial q_{1_2}} &= 0 \\
\frac{\partial z}{\partial q_{2_1}} &= 0 \\
\frac{\partial z}{\partial q_{2_2}} &= 0
\end{aligned} . \tag{68}$$

Building Jacobian matrix $J(q_{1_1}, q_{1_2}, q_{2_1}, q_{2_2})$ will obtain the partial derivative and derivative together:

$$J(q_{1_1}, q_{1_2}, q_{2_1}, q_{2_2}) = \begin{bmatrix} \frac{\partial x}{\partial q_{1_1}} & \frac{\partial x}{\partial q_{1_2}} & \frac{\partial x}{\partial q_{2_1}} & \frac{\partial x}{\partial q_{2_2}} \\ \frac{\partial y}{\partial q_{1_1}} & \frac{\partial y}{\partial q_{1_2}} & \frac{\partial y}{\partial q_{2_1}} & \frac{\partial y}{\partial q_{2_2}} \\ \frac{\partial z}{\partial q_{1_1}} & \frac{\partial z}{\partial q_{1_2}} & \frac{\partial z}{\partial q_{2_1}} & \frac{\partial z}{\partial q_{2_2}} \end{bmatrix}. \tag{69}$$

According to the Jacobian matrix $J(q_{1_1}, q_{1_2}, q_{2_1}, q_{2_2})$, joint speed $[\dot{x}, \dot{y}, \dot{z}]^T$ and the end executor speed $[\dot{q}_{1_1}, \dot{q}_{1_2}, \dot{q}_{2_1}, \dot{q}_{2_2}]^T$ can be expressed by the following formula:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J(q_{1_1}, q_{1_2}, q_{2_1}, q_{2_2}) \begin{bmatrix} \dot{q}_{1_1} \\ \dot{q}_{1_2} \\ \dot{q}_{2_1} \\ \dot{q}_{2_2} \end{bmatrix}. \tag{70}$$

The relationship between the joint velocity and the end-effector velocity is obtained by the inverse operation:

$$\begin{bmatrix} \dot{q}_{1_1} \\ \dot{q}_{1_2} \\ \dot{q}_{2_1} \\ \dot{q}_{2_2} \end{bmatrix} = J(q_{1_1}, q_{1_2}, q_{2_1}, q_{2_2})^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \tag{71}$$

4.2. Simulation Setup

When the value of n is 2 in Equation (23), the expressions for $D(q)$, $C(q,\dot{q})$, and $G(q)$ can be formulated in the following manner:

$$\begin{aligned} D(q) &= \begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 \end{bmatrix} \\ C(q,\dot{q}) &= \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}, \\ G(q) &= \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \end{aligned} \tag{72}$$

where

$$\begin{aligned} D_1 &= m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos q_2 + J_1 + J_2 \\ D_2 &= D_3 = m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos q_2 + J_2 \\ D_4 &= m_2 l_{c2}^2 + J_2 \\ C_1 &= -m_2 l_1 l_{c2} \dot{q}_2 \sin q_2 \\ C_2 &= -m_2 l_1 l_{c2} \dot{q}_1 \sin q_2 - m_2 l_1 l_{c2} \dot{q}_2 \sin q_2 \\ C_3 &= m_2 l_1 l_{c2} \dot{q}_1 \sin q_2 \\ C_4 &= 0 \\ G_1 &= m_1 l_{c2} g \cos q_1 + m_2 l_1 g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2) \\ G_2 &= m_2 l_{c2} g \cos(q_1 + q_2) \end{aligned} \tag{73}$$

In the given system, the parameters $m_1 = m_2 = 2$ kg and $l_1 = l_2 = 0.5$ m represent the mass and length of each link ($i = 1, 2$), respectively. The term l_{ci} $(i = 1, 2)$ represents the measurement of the separation between the i-1th joint and the center of mass of the ith link. The rotational inertia $J_1 = J_2 = 0.125$ kgm^2 is computed for each link.

The reference trajectory is specified as $q_d = [\sin(0.3t^2), 2\cos(0.3t^2)]^T$. To ensure satisfactory transient performance, the output constraints ξ_{1i} and ξ_{2i} are specified as follows:

$$\begin{aligned} \xi_{11} = \xi_{12} &= -\left(0.92e^{-t} + 0.08\right) \\ \xi_{21} = \xi_{22} &= 0.92e^{-t} + 0.08 \end{aligned} \tag{74}$$

To design the controller, the parameters can be assigned as follows: $k_{1i} = k_{2i} = k_{3i} = 2$, $k_{a1} = k_{a2} = 0.5$, $k_{4i} = k_{5i} = k_{6i} = 2$, and $\epsilon_i = 0.01$. The adaptive laws parameters in Equation (49) are modified to $Q_1 = \text{diag}\{0.16\}$ and $Q_2 = \text{diag}\{0.3\}$.

The simulated fault model can be inferred from Equation (17), where $m_i = i = 2$. At the beginning of the normal operation of robot, until the fifth second, a sudden actuator failure occurs in the robot. The duration of the fault is 5 s. Specifically, the first and second joints of the robot experience an actuator failure, while the remaining actuators continue to function normally. This can be expressed as $\sigma_{12} = \sigma_{22} = 0$.

4.3. Simulation Results

By selecting the control design parameters and initial controller parameters, the plots shown in Figures 3–7 are obtained. From Figure 3, it can be observed that the tracking trajectories of the two joint angles closely follow the desired trajectories during the first five seconds. However, at the fifth second, when the actuator failure occurs, it is evident that the tracking trajectory of the first joint angle deviates significantly from the desired trajectory. Overall, the tracking trajectories show a slightly lower level of fit to the desired trajectories when there is an actuator failure compared to when there is no failure. Figure 4 clearly demonstrates that the tracking error e_i converges to a small value close to zero within 2 s. It can also be observed that there is a noticeable increase in tracking error for

both joints when the actuator failure suddenly occurs, and the tracking error with actuator failure is slightly larger than that without failure. During operation, we can ensure that the tracking error e_i remains within the range of $\xi_{1i} < e_i < \xi_{2i}$, thus guaranteeing the predefined transient performance. The torque τ_i is shown in Figure 5. At the fifth second, both τ_1 and τ_2 experience a significant jump. The adaptive parameters of the system are illustrated in Figures 6 and 7.

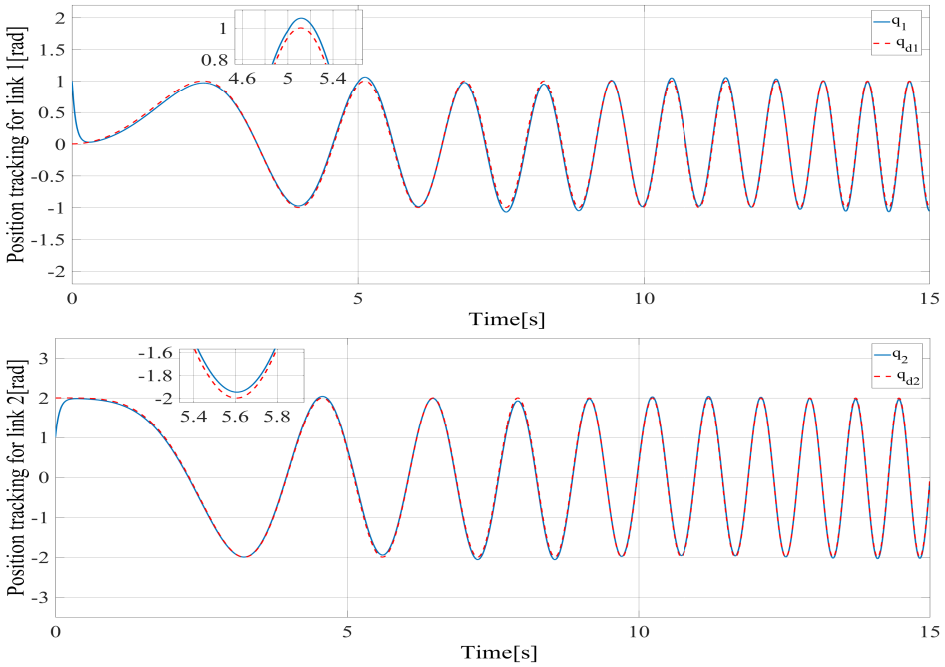


Figure 3. Performance tracking of the joint positions q_1 and q_2.

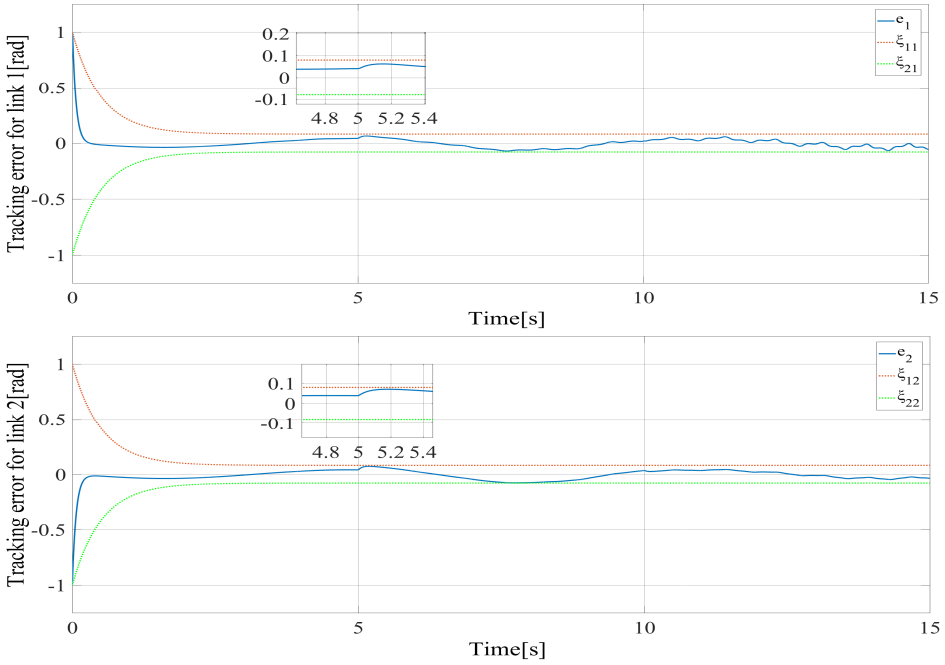


Figure 4. Performance tracking of the positional errors e_1 and e_2.

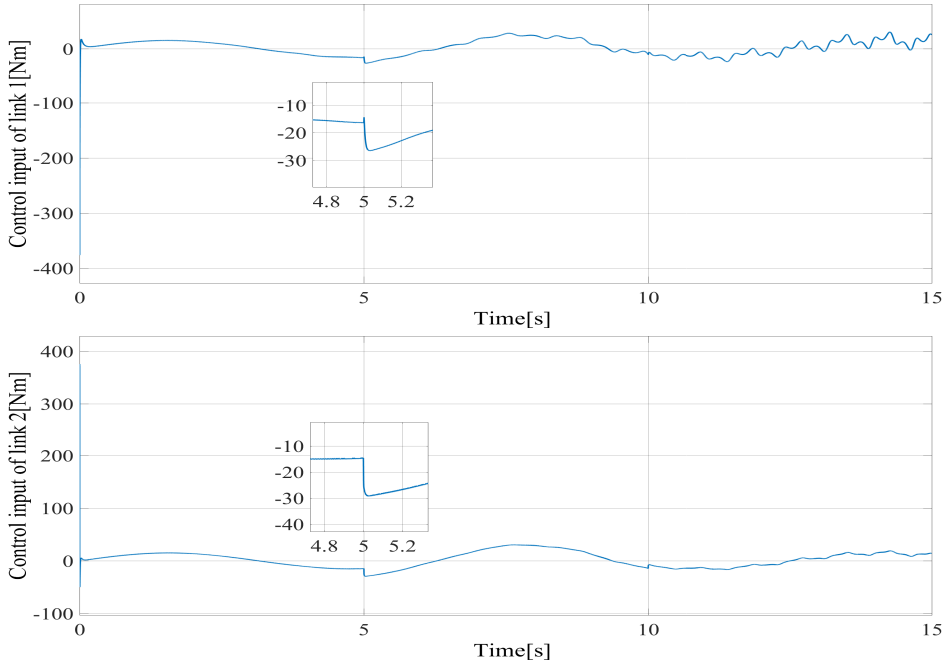


Figure 5. Control input τ_1 and τ_2.

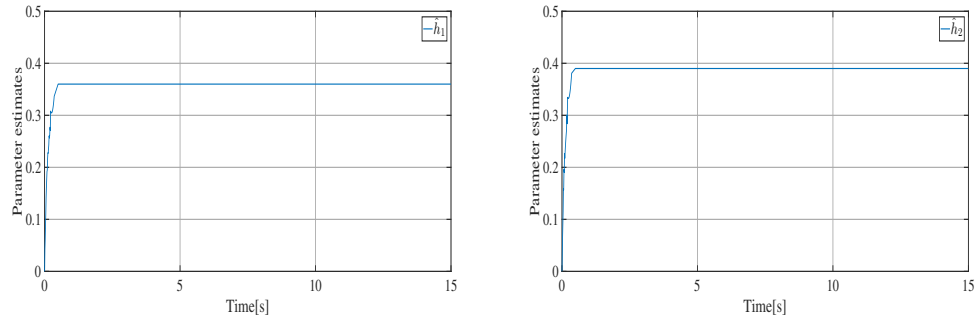


Figure 6. Parameter estimates $\hat{h}_1$ and $\hat{h}_2$.

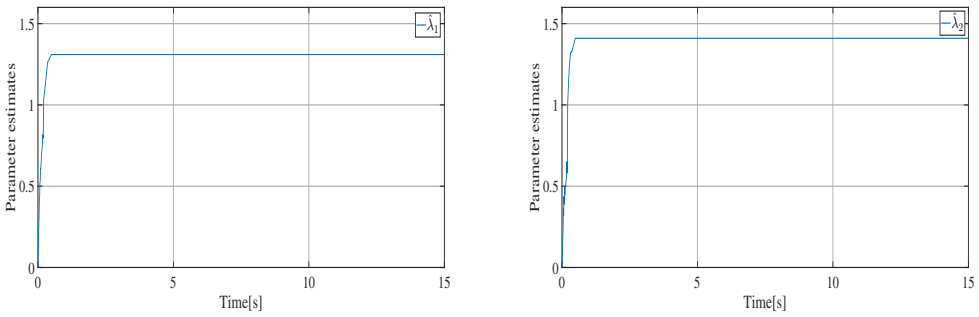


Figure 7. Parameter estimates $\hat{\lambda}_1$ and $\hat{\lambda}_2$.

5. Conclusions

This paper presents an adaptive control approach to address the challenge of actuator faults in unknown robots with predefined output constraints. A decoupling-based method is introduced to address actuator faults, and an adaptive rule is proposed to handle system delays resulting from the actuator fault. The unknown system dynamics are approximated using RBFNN, and a neural control scheme is designed to achieve tracking. Unlike conventional fixed-time algorithms, this approach limits tracking errors to predefined ranges instead of uncertain convergence domains.

Author Contributions: Conceptualization, G.L.; methodology, W.Y.; software, S.Z.; validation, G.L. and X.W.; formal analysis, S.Z.; writing—original draft preparation, S.Z.; writing—review and editing, F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Tertiary Education Scientific research project of Guangzhou Municipal Education Bureau [No.202235364], and the Guangzhou Higher Education Teaching Quality and teaching construction reform project [No.2022JXMS016], and the Special projects in key fields of colleges and universities in Guangdong Province [No.2021ZDZX1109, No.2022ZDZX1070], and the National Natural Science Foundation of China [No.6210021076], and the Guangzhou Municipal Science and Technology Project [No.202201010381].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Dai, S.L.; Lu, K.; Jin, X. Fixed-Time Formation Control of Unicycle-Type Mobile Robots with Visibility and Performance Constraints. *IEEE Trans. Ind. Electron.* **2020**, *68*, 12615–12625. [CrossRef]
2. Lu, K.; Liu, Z.; Lai, G.; Chen, C.L.P.; Zhang, Y. Adaptive Consensus Tracking Control of Uncertain Nonlinear Multiagent Systems With Predefined Accuracy. *IEEE Trans. Cybern.* **2021**, *51*, 405–415. [CrossRef]
3. Li, W.; Song, C.; Li, Z. An Accelerated Recurrent Neural Network for Visual Servo Control of a Robotic Flexible Endoscope with Joint Limit Constraint. *IEEE Trans. Ind. Electron.* **2019**, *67*, 10787–10797. [CrossRef]
4. Liu, Y.; Gong, M.; Liu, L.; Tong, S.; Chen, C. Fuzzy Observer Constraint Based on Adaptive Control for Uncertain Nonlinear MIMO Systems With Time-Varying State Constraints. *IEEE Trans. Cybern.* **2021**, *51*, 1380–1389. [CrossRef]
5. Chen, J.; Hua, C. Adaptive Full-State-Constrained Control of Nonlinear Systems With Deferred Constraints Based on Nonbarrier Lyapunov Function Method. *IEEE Trans. Cybern.* **2020**, *52*, 7634–7642. [CrossRef]
6. Sun, T.; Pei, H.; Pan, Y.; Zhou, H.; Zhang, C. Neural network-based sliding mode adaptive control for robot manipulators. *Neurocomputing* **2011**, *74*, 2377–2384. [CrossRef]
7. Jiang, Y.; Wang, Y.; Miao, Z.; Na, J.; Zhao, Z.; Yang, C. Composite-Learning-Based Adaptive Neural Control for Dual-Arm Robots With Relative Motion. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1010–1021. [CrossRef] [PubMed]
8. Zhang, S.; Dong, Y.; Ouyang, Y.; Yin, Z.; Peng, K. Adaptive Neural Control for Robotic Manipulators With Output Constraints and Uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5554–5564. [CrossRef] [PubMed]
9. Yang, C.; Huang, D.; He, W.; Cheng, L. Neural Control of Robot Manipulators With Trajectory Tracking Constraints and Input Saturation. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4231–4242. [CrossRef] [PubMed]
10. Yang, C.; Peng, G.; Li, Y.; Cui, R.; Li, Z. Neural Networks Enhanced Adaptive Admittance Control of Optimized Robot-Environment Interaction. *IEEE Trans. Cybern.* **2018**, *49*, 2568–2579. [CrossRef]
11. Zhang, F.; Huang, P. Fuzzy-based Adaptive Super-Twisting Sliding-Mode Control for a Maneuverable Tethered Space Net Robot. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 1739–1749. [CrossRef]
12. Han, J.; Yang, S.; Xia, L.; Chen, Y.H. Deterministic Adaptive Robust Control With a Novel Optimal Gain Design Approach for a Fuzzy 2-DOF Lower Limb Exoskeleton Robot System. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 2373–2387. [CrossRef]
13. Pan, Y.; Du, P.; Xue, H.; Lam, H.K. Singularity-Free Fixed-Time Fuzzy Control for Robotic Systems With User-Defined Performance. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 2388–2398. [CrossRef]
14. Lewis, F.L.; Jagannathan, S.; Yesildirek, A. *Neural Network Control of Robot Manipulators and Nonlinear Systems*; CRC Press: Boca Raton, FL, USA, 1999.
15. Qiu, J.; Wang, T.; Sun, K.; Rudas, I.J.; Gao, H. Disturbance Observer-Based Adaptive Fuzzy Control for Strict-Feedback Nonlinear Systems with Finite-Time Prescribed Performance. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 1175–1184. [CrossRef]
16. Zhu, B.; Wang, Y.; Zhang, H.; Xie, X. Fuzzy Functional Observer-based Finite-Time Adaptive Sliding Mode Control for Nonlinear Systems with Matched Uncertainties. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 918–932. [CrossRef]
17. Sui, S.; Xu, H.; Tong, S.; Chen, C.L.P. Prescribed Performance Fuzzy Adaptive Output Feedback Control for Nonlinear MIMO Systems in a Finite Time. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 3633–3644. [CrossRef]
18. Zhu, C.; Jiang, Y.; Yang, C. Fixed-Time Neural Control of Robot Manipulator With Global Stability and Guaranteed Transient Performance. *IEEE Trans. Ind. Electron.* **2023**, *70*, 803–812. [CrossRef]
19. Chakravarty, A.; Nizami, T.K.; Kar, I.; Mahanta, C. Adaptive Compensation of Actuator Failures using Multiple Models. *IFAC-PapersOnLine* **2017**, *50*, 10350–10356. [CrossRef]
20. Liu, L.; Shi, K.; Yuan, X.; Li, Q. Multiple model-based fault-tolerant control system for distributed drive electric vehicle. *J. Braz. Soc. Mech. Sci. Eng.* **2019**, *41*, 531. [CrossRef]

21. Amini, N.; Zhu, Q. Fault detection and diagnosis with a novel source-aware autoencoder and deep residual neural network. *Neurocomputing* **2022**, *488*, 618–633. [CrossRef]
22. Zhong, Y.; Zhang, W.; Zhang, Y.; Zuo, J.; Zhan, H. Sensor Fault Detection and Diagnosis for an Unmanned Quadrotor Helicopter. *J. Intell. Robot. Syst.* **2019**, *96*, 555–572. [CrossRef]
23. Samir, Z.; Hemza, M.; Abderrahmen, B.; Ali, D. Actuator fault tolerant control using adaptive RBFNN fuzzy sliding mode controller for coaxial octorotor UAV. *ISA Trans.* **2018**, *80*, S0019057818302234.
24. Hu, Q.; Shao, X.; Guo, L. Adaptive Fault-Tolerant Attitude Tracking Control of Spacecraft With Prescribed Performance. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 331–341. [CrossRef]
25. Mao, Z.; Yan, X.G.; Jiang, B.; Chen, M. Adaptive Fault-Tolerant Sliding-Mode Control for High-Speed Trains With Actuator Faults and Uncertainties. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 2449–2460. [CrossRef]
26. Qi, H.; Shi, Y.; Li, S.; Tian, Y.; Yu, D.L.; Gomm, J.B. Fault tolerant control for nonlinear systems using sliding mode and adaptive neural network estimator. *Soft Comput.* **2020**, *24*, 11535–11544. [CrossRef]
27. Liu, Y.J.; Zeng, Q.; Tong, S.; Chen, C.; Liu, L. Actuator Failure Compensation-Based Adaptive Control of Active Suspension Systems With Prescribed Performance. *IEEE Trans. Ind. Electron.* **2020**, *67*, 7044–7053. [CrossRef]
28. Wang, J.; Liu, Z.; Chen, C.L.P.; Zhang, Y.; Lai, G. Extended dimension fuzzy adaptive control for nonlinear uncertain stochastic systems with actuator constraints. *Nonlinear Dyn.* **2019**, *98*, 1315–1329. [CrossRef]
29. Ouyang, H.; Lin, Y. Adaptive Fault-Tolerant Control and Performance Recovery Against Actuator Failures With Deferred Actuator Replacement. *IEEE Trans. Autom. Control* **2021**, *66*, 3810–3817. [CrossRef]
30. Tao, G.; Chen, S.; Joshi, S.M. An adaptive actuator failure compensation controller using output feedback. *Autom. Control IEEE Trans.* **2002**, *47*, 506–511. [CrossRef]
31. Lai, G.; Wen, C.; Liu, Z.; Zhang, Y.; Chen, C.P.; Xie, S. Adaptive compensation for infinite number of actuator failures/faults using output feedback control. *Inf. Sci.* **2017**, *399*, 1–12. [CrossRef]
32. Tang, X.; Tao, G. An adaptive nonlinear output feedback controller using dynamic bounding with an aircraft control application. *Int. J. Adapt. Control Signal Process.* **2009**, *23*, 609–639. [CrossRef]
33. Tang, X.; Gang, T.; Joshi, S.M. Adaptive actuator failure compensation for parametric strict feedback systems and an aircraft application. *Automatica* **2003**, *39*, 1975–1982. [CrossRef]
34. Wang, W.; Wen, C. *Adaptive Actuator Failure Compensation Control of Uncertain Nonlinear Systems with Guaranteed Transient Performance*; Automatica Oxford: Oxford, UK, 2010.
35. Rugthum, T.; Tao, G. An adaptive actuator failure compensation scheme for a cooperative manipulator system. *Robotica* **2016**, *34*, 1529–1552. [CrossRef]
36. Rugthum, T. Dynamic structure adaptive actuator failure compensation scheme for robotic systems. In Proceedings of the 2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST), Manado, Indonesia, 26–27 October 2018; pp. 1–4.
37. Kececi, E.F.; Tang, X.; Gang, T. Adaptive actuator failure compensation for redundant manipulators. *Robotica* **2009**, *27*, 19–28. [CrossRef]
38. Zhang, W.; Tan, M.; Hou, Z.; Cheng, L. Tracking Control of a Closed-Chain Five-Bar Robot with Two Degrees of Freedom by Integration of an Approximation-Based Approach and Mechanical Design. *IEEE Trans. Syst. Man, Cybern. Part B Cybern.* **2012**, *42*, 1470–1479.
39. Huang, Y.; Jia, Y. Adaptive Fixed-Time Six-DOF Tracking Control for Noncooperative Spacecraft Fly-Around Mission. *IEEE Trans. Control. Syst. Technol.* **2018**, *27*, 1796–1804. [CrossRef]
40. Zuo, Z. *Nonsingular Fixed-Time Consensus Tracking for Second-Order Multi-Agent Networks*; Pergamon Press, Inc.: Oxford, UK, 2015.
41. Huang, J.T. Global Tracking Control of Strict-Feedback Systems Using Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1714–1725. [CrossRef]
42. Zhang, L.; Wang, Y.; Hou, Y.; Li, H. Fixed-Time Sliding Mode Control for Uncertain Robot Manipulators. *IEEE Access* **2019**, *7*, 149750–149763. [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.