

Article

Fixed-Time Adaptive Neural Network-Based Trajectory Tracking Control for Workspace Manipulators

Xiaofei Chen ^{1,2}, Han Zhao ¹, Shengchao Zhen ¹, Xiaoxiao Liu ²  and Jinsi Zhang ^{2,*}¹ School of Mechanical Engineering, Hefei University of Technology, Hefei 230000, China; 2020030006@mail.hfut.edu.cn (X.C.)² School of Electrical and Optoelectronic Engineering, West Anhui University, Liu'an 237000, China; 42000002@wxc.edu.cn

* Correspondence: 42000007@wxc.edu.cn

Abstract: This paper proposes a novel neural network-based control algorithm with fixed-time performance constraints for manipulator systems in workspaces. The algorithm efficiently controls the manipulator's trajectory tracking by tuning a preset performance function, thereby optimizing both speed and accuracy within a fixed timeframe. Initially, a tangent-type error transformation, applied through homogeneous embryonic transformation, ensures rapid convergence of tracking errors to a specific region. Subsequently, integrating a predetermined control strategy into the fixed-time stability framework ensures the system's state reaches a defined boundary within a finite period. Lastly, neural networks are employed to approximate dynamic parameters and adjust the controller, achieving optimal parameter approximation and significantly enhancing trajectory tracking robustness. Simulation analyses and comparisons confirm the controller's effectiveness and superiority in enhancing both the transient and steady-state performance of the control system.

Keywords: manipulator; fixed time; preset performance; neural network; trajectory tracking



Citation: Chen, X.; Zhao, H.; Zhen, S.; Liu, X.; Zhang, J. Fixed-Time Adaptive Neural Network-Based Trajectory Tracking Control for Workspace Manipulators. *Actuators* **2024**, *13*, 252. <https://doi.org/10.3390/act13070252>

Academic Editor: Zhuming Bi

Received: 23 May 2024

Revised: 28 June 2024

Accepted: 29 June 2024

Published: 1 July 2024



Copyright: © 2024 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

As modern industry rapidly evolves, manipulators are increasingly utilized in industrial production. Owing to their high efficiency, flexibility, and precision, manipulators play a vital role in industrial production. However, the control of manipulators remains a challenging and keenly studied area due to their highly nonlinear, strongly coupled, and time-varying nature [1–3].

Addressing the manipulator's trajectory tracking control, scholars globally have extensively researched and proposed methods including proportion integration differentiation (PID) control [4–6], inverse-step control [7,8], and sliding mode control [9,10]. Despite their simplicity and usability, these methods struggle to fulfill high accuracy and robustness requirements. To address these shortcomings, researchers have introduced additional methods like adaptive control [11–14], robust control [15–18], and fuzzy control [19–22]. While these methods improve control performance and handle system uncertainty, their slow convergence speed fails to meet rapid response requirements.

Recent years have seen the proposal of new control schemes to meet the growing demands for manipulator control accuracy in the industry. Within these control schemes, certain studies specifically address the manipulator system's uncertainty. For instance, Ref. [23] describes combining adaptive backstepping control with fuzzy logic systems' function approximation capabilities to create controllers for input-saturated manipulator systems. Moreover, Ref. [24] addresses how environmental or inherent conditions may limit a manipulator's position and angular velocity, achieving indirect system state constraints through position error limitations using the obstacle Lyapunov function. Other studies have focused on addressing issues within traditional control schemes. The article [25] introduces an enhanced design method for a robust, optimal, flexible jointed manipulator

interference observer, aiming to maximize interference suppression and ensure optimal trajectory tracking. While the paper primarily improves the steady-state accuracy of flexible jointed manipulator position tracking, it overlooks in-depth analysis of transient response and output constraints. Study [26] employs a nonlinear filter with a self-compensation feature to correct boundary layer errors and mitigate filter inaccuracies. While this scheme enhances control performance, the nonlinear filter significantly complicates the overall control system.

To further improve the control performance of manipulator systems, preset performance control [27–29] and fixed-time control [30,31] have received much attention. The fixed-time control ensures tracking error convergence to a steady state within a set period, yet it overlooks transient performance constraints. The preset performance control is able to satisfy both the transient and steady-state performance requirements of the manipulators, but it cannot explicitly provide the time at which the system enters the steady state. The control scheme proposed in study [32] ensures both transient and steady state performance of the system but does not allow the tracking error to converge within a predefined performance boundary in a predefined time. Study [33] merges finite time with predefined performance control, offering a trajectory tracking scheme. However, its convergence rate, tied to a fixed stopping time parameter, limits flexibility and transient performance enhancement.

Despite the extensive research, controller design frequently relies on fully available robot dynamics models, which are seldom accessible in real-world scenarios. Consequently, exploring advanced control strategies to address model uncertainties becomes critically important. Neural networks (NN) are renowned and utilized for their superior function approximation abilities, particularly in simplifying the modeling of nonlinear systems [34,35]. This implies new solution avenues for manipulator control issues, especially those lacking precise dynamic models, through the robust modeling and approximation prowess of neural networks. Employing neural networks comprehensively for control compensates for traditional model deficiencies, enhances system stability and accuracy, and facilitates the broader application and development of manipulator systems across various fields.

Building on the previous discussion, this study integrates a pre-programmed performance control strategy within a fixed-time stability framework and employs adaptive neural network control to devise a manipulator control strategy for superior performance under stringent constraints. Our study's achievement of quicker response and heightened precision not only boosts the manipulators' operational performance in intricate scenarios but also its adaptability and flexibility, all while maintaining constraint compliance and system safety, marking significant theoretical and practical advancements. Relative to existing methods, this paper's primary contributions are as follows:

1. The homogeneous embryo transformation enables tangent-type error transformation, ensuring tracking error convergence to a specific region and preventing "performance spikes" seen in traditional control methods;
2. The integration of the preset performance control strategy into the fixed-time stability theoretical framework not only ensures that the state converges to a predetermined region within a predetermined time, but also further adjusts the convergence speed and accuracy to achieve better control performance;
3. The use of neural networks to approximate the unknown dynamical parameters and compensate the prediction results to the controller improves the robustness and adaptive ability of the controller without the need of known uncertainty boundaries of the system.

This paper is structured as follows: Section 2 introduces the system model and performance function. Section 3 details the design of a neural network controller with fixed-time preset performance. Section 4 presents simulation validation to demonstrate the algorithm's effectiveness. Section 5 concludes the paper and outlines future research directions.

2. System Modeling and Problem Formulation

2.1. Coordinate Transformation

To represent the model in the workspace, the manipulator joint coordinates (ϕ_1, ϕ_2) need to be transformed into the end-effector Cartesian coordinates (x_1, x_2) .

The schematic structure of the two-degree-of-freedom manipulator is shown in Figure 1, and modeling of the manipulator in Figure 1 yields

$$x_1 = l_1 \cos \phi_1 + l_2 \cos(\phi_1 + \phi_2) \quad (1)$$

$$x_2 = l_1 \sin \phi_1 + l_2 \sin(\phi_1 + \phi_2) \quad (2)$$

Combining Equations (1) and (2), we obtain

$$x_1^2 + x_2^2 = l_1^2 + l_2^2 + 2l_1l_2 \cos \phi_2 \quad (3)$$

Then

$$\phi_2 = \arccos\left(\frac{x_1^2 + x_2^2 - l_1^2 - l_2^2}{2l_1l_2}\right) \quad (4)$$

Let $q_1 = \arctan \frac{x_2}{x_1}$ and $q_2 = \arccos \frac{x_1^2 + x_2^2 + l_1^2 - l_2^2}{2l_1 \sqrt{x_1^2 + x_2^2}}$ and the following can be obtained:

$$\phi_1 = \begin{cases} q_1 - q_2, \phi_2 > 0 \\ q_1 + q_2, \phi_2 \leq 0 \end{cases} \quad (5)$$

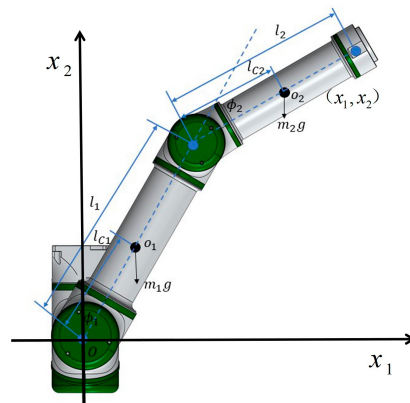


Figure 1. Schematic diagram of 2-DOF manipulator.

2.2. Neural Network Modeling of Manipulators

Consider a rigid 2-degree-of-freedom manipulator as the controlled object. The manipulator's operation is assumed to be related to the end-effector's spatial position. Let $x \in R^n$ represent the end-effector's location and orientation within the workspace. The following is an expression of the manipulator's dynamic properties in the workspace:

$$H_x(\phi)\ddot{x} + C_x(\phi, \dot{\phi})\dot{x} + g_x(\phi) = \tau_x \quad (6)$$

where $H_x(\phi)$ is the $n \times n$ order rotational inertia matrix, $H_x(\phi) = J^{-T}(\phi)H(\phi)J^{-1}(\phi)$; $C_x(\phi, \dot{\phi})$ is the $n \times n$ order matrix of the Göttinger force and centrifugal force, $C_x(\phi, \dot{\phi}) = J^{-T}(\phi)(C(\phi, \dot{\phi}) - H(\phi)J^{-1}(\phi)\dot{J}(\phi))J^{-1}(\phi)$; $g_x(\phi)$ is the $n \times 1$ order gravity vector, $g_x(\phi) = J^{-T}(\phi)g(\phi)$; and $\tau_x \in R^n$ is the vector of joint torques exerted by the actuator. $J(\phi) \in R^{n \times n}$ is the Jacobi matrix determined by the structure of the system, which is assumed to be non-singular in the bounded workspace.

The dynamic equations of the manipulator have the following properties:

Property 1. The inertia matrix $H_x(\phi)$ is symmetrically positive definite.

Property 2. If $C_x(\phi, \dot{\phi})$ is defined via the Christoffel sign rule, the matrix $\dot{H}_x(\phi) - 2C_x(\phi, \dot{\phi})$ is skew-symmetric.

The matrices H_x , C_x , and g_x contain parametric uncertainties, which can negatively impact the control (6). Therefore, neural networks can be employed to handle these unknown dynamics.

Definition 1. A class of Radial Basis Function Neural Networks (RBFNN) is commonly used to estimate unknown continuous functions as follows [36]:

$$f_i(\Xi) = \hat{W}_i^T S_i(\Xi) + \bar{\varepsilon}_i(\Xi) \quad i = 1, 2, \dots, n \quad (7)$$

where $\hat{W}_i^* \in R^{l \times n}$ is the ideal weight matrix, $S_i(\Xi) \in R^{l \times n}$ is the Gaussian radial basis function matrix, and $\bar{\varepsilon}_i(\Xi) = [\bar{\varepsilon}_1(\Xi), \dots, \bar{\varepsilon}_n(\Xi)]^T$ is the unknown estimated error column vector. The index i represents the i -th neuron in the RBFNN and ranges from 1 to n , where n is the total number of neurons. $\hat{W}_i^*$ is unknown and its estimate $\hat{W}_i$ is used to design the controller and has $\tilde{W}_i = \hat{W}_i - W_i^*$, denoting the error value of the weight matrix. $\bar{\varepsilon}_i(\Xi)$ satisfies $\bar{\varepsilon}_i(\Xi) \leq \lambda < \infty$.

The radial basis element s_j of $S_i(\Xi)$ is provided by the following equation:

$$s_j(\Xi) = \exp\left(\frac{-(\Xi - \mu_j)^T(\Xi - \mu_j)}{\sigma_j^2}\right) \quad (8)$$

where μ_j is the center of the receptive field, and σ_j is the width of the Gaussian function.

Since $H_x(\phi)$ and $g_x(\phi)$ are functions of ϕ only, they can be modeled using a static neural network.

$$H_{xkj}(\phi) = \sum_l \theta_{kjl} \zeta_{kjl}(\phi) + \varepsilon_{hkj}(\phi) = \theta_{kj}^T \zeta_{kj}(\phi) + \varepsilon_{hkj}(\phi) \quad (9)$$

$$g_{xk}(\phi) = \sum_l m_{kl} \zeta_{kl}(\phi) + \varepsilon_{gk}(\phi) = m_k^T \zeta_k(\phi) + \varepsilon_{gk}(\phi) \quad (10)$$

where the weights of the neural network are represented by $\theta_{kjl}, m_{kl} \in R$, the radial basis function with respect to the input vector ϕ is represented by $\zeta_{kjl}(\phi), \zeta_{kl}(\phi) \in R$, and the modeling errors for $H_{xkj}(\phi)$ and $g_{xk}(\phi)$, respectively, are represented by $\varepsilon_{mkj}(\phi), \varepsilon_{gk}(\phi) \in R$, and they are presumptively limited.

The neural network model for $C_x(\phi, \dot{\phi})$ is based on a dynamic neural network with inputs ϕ and $\dot{\phi}$ for $C_{xkj}(\phi, \dot{\phi})$,

$$C_{xkj}(\phi, \dot{\phi}) = \sum_l n_{kjl} \zeta_{kjl}(\Phi) + \varepsilon_{ckj}(\Phi) = n_{kj}^T \zeta_{kj}(\Phi) + \varepsilon_{ckj}(\Phi) \quad (11)$$

where $\Phi = [\phi^T \dot{\phi}^T]^T \in R^{2n}$; $n_{kjl} \in R$ are the weights; $\zeta_{kjl}(\Phi) \in R$ is the radial basis function whose inputs are the vectors Φ ; and $\varepsilon_{ckj}(\Phi)$ is the modeling error of $C_{xkj}(\phi, \dot{\phi})$ and is assumed to be bounded as well.

2.3. Fixed-Time Preset Performance Function

This paper employs a performance-constrained control method that limits the tracking error within a preset range using an exponentially decaying performance function. However, the traditional exponential decay performance function only ensures convergence to the preset range in infinite time, but not within a specified time. To achieve better tracking

performance within a specified time, a specified-time preconditioned performance function is introduced, defined as follows:

$$\beta_i(t) = \begin{cases} \left(\frac{T_i-t}{T_i}\right)^{\frac{\rho_i}{1-\rho_i}} (\beta_i^0 - \beta_i^\infty) + \beta_i^\infty & 0 \leq t \leq T_i \\ \beta_i^\infty & t > T_i \end{cases} \quad (12)$$

where the performance function constants β_i^0 and β_i^∞ have value domains of $\beta_i^0 > \beta_i^\infty > 0$ and $\rho_i = L_1 - L_2 \cos(t\pi/T_i)$. T_i are freely set time parameters. Here, L_1 represents the maximum value of the performance function, and L_2 represents the amplitude of the oscillation around the mean value defined by L_1 .

Based on the proposed performance function, the boundary constraint of the position tracking error variable is as follows:

$$-\eta_i \beta_i(t) < e_i < \bar{\eta}_i \beta_i(t) \quad (13)$$

where $-\eta_i$ and $\bar{\eta}_i$ are constants. The initial value of Equation (13) is as follows:

$$-\eta_i \beta_i(0) < e_i < \bar{\eta}_i \beta_i(0) \quad (14)$$

Since the performance function (12) cannot be used directly to design the controller, an error transformation function is needed to relate the tracking error to the specified time performance function, consider the error transformation function $\psi(\cdot)$.

$$\psi(\varepsilon_i) = \frac{e_i(t)}{\beta_i(t)} = \frac{2\eta_i}{\pi} \arctan(\varepsilon_i) \quad (15)$$

where ε_i is the transformed position tracking error, and function $\psi(\varepsilon_i)$ has the following properties:

- (1) The function $\psi(\varepsilon_i)$ is strictly growing and smooth;
- (2) $-\eta_i < \psi(\varepsilon_i) < \bar{\eta}_i$;
- (3) $\lim_{\varepsilon_i \rightarrow -\infty} \psi(\varepsilon_i) = -\eta_i$, $\lim_{\varepsilon_i \rightarrow +\infty} \psi(\varepsilon_i) = \bar{\eta}_i$.

After the above transformation, the unconstrained tracking error e_i is transformed into a constrained error variable ε_i , which can reduce the difficulty of controller design.

3. Controller Design

The control framework diagram for trajectory tracking control of a workspace manipulator based on a fixed-time adaptive neural network is illustrated in Figure 2. The control system begins with an error equivalent transformation using a homoembryonic approach, which ensures rapid convergence of tracking errors. This is followed by the application of fixed-time predetermined performance constraints. Adaptive neural networks are then employed to approximate dynamic parameters and adjust the controller. The dynamic model of the manipulator is controlled to achieve precise trajectory tracking within a fixed timeframe, optimizing both speed and accuracy. The workspace manipulator's state is dynamically adjusted to meet the desired performance criteria.

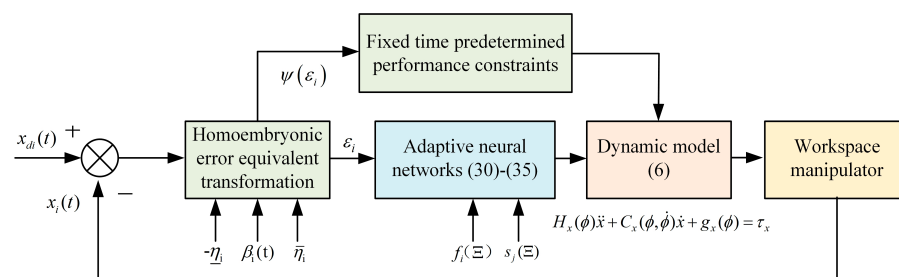


Figure 2. Control scheme block diagram.

3.1. Homoembryonic Error Equivalent Transformation

Let $x_d(t)$ be the ideal trajectory in the workspace, then $\dot{x}_d(t)$ and $\ddot{x}_d(t)$ are the ideal velocity and acceleration, respectively. Before controller design, the tracking error signal of the robot manipulator can be defined as follows:

$$e_i(t) = x_i(t) - x_{di}(t) \quad (16)$$

In order to make the actual trajectory $x(t)$ of the manipulator track the desired signal $x_d(t)$, and to make the system obtain better steady-state tracking performance, this paper is based on the theory of fixed-time stability, and, at the same time, combined with the predetermined performance control, the controller constraint strategy is designed to satisfy the following conditions:

- (1) The system's adjustment time must be less than 1 s under any operating conditions and reference signal changes;
- (2) The steady state tracking error is less than 0.03;
- (3) The tracking error of the system is limited to the upper and lower bounds of the preset performance function.

To achieve the aforementioned control objectives, an error transformation is introduced. It converts the original nonlinear system with constrained tracking error behavior into an equivalent unconstrained system.

$$\varepsilon_i(t) = \psi^{-1} \frac{e_i(t)}{\beta_i(t)} \triangleq Z \frac{e_i(t)}{\beta_i(t)} \quad (17)$$

where $\varepsilon_i(t) \in R$ is the transformed error vector, $\psi(\varepsilon_i) = [\psi(\varepsilon_1), \dots, \psi(\varepsilon_n)]^T$.

$$\psi(\varepsilon_i) = \frac{2\eta_i}{\pi} \arctan(\varepsilon_i) \quad (18)$$

$$\varepsilon_i(t) = \tan \frac{\pi \psi(\varepsilon_i)}{2\eta_i} \quad (19)$$

where η_i is a design scalar constant which, together with β_i , imposes behavioral bounds on the system output tracking error, β_i , as shown in Equation (12). As can be seen, the transformation (17) is one-to-one, meaning that if the new error boundaries of $\varepsilon_i(t)$ are fulfilled, a fixed-time prescribed performance control of the tracking error $e_i(t)$ may be obtained.

Additionally, the following equation provides the first- and second-order derivatives of $\psi(\varepsilon_i)$ for the controller design that comes next, derived from Equation (18):

$$\dot{\psi}(\varepsilon_i) = \frac{2\dot{\eta}_i}{\pi} \arctan(\varepsilon_i) + \frac{2\eta_i}{\pi} \frac{\dot{\varepsilon}_i}{1 + \varepsilon_i^2} \quad (20)$$

$$\ddot{\psi}(\varepsilon_i) = \frac{2\ddot{\eta}_i}{\pi} \arctan \varepsilon_i + \frac{4\dot{\eta}_i}{\pi} \frac{\dot{\varepsilon}_i}{1 + \varepsilon_i^2} + \frac{2\eta_i}{\pi} \frac{\ddot{\varepsilon}_i(1 + \varepsilon_i^2) - 2\varepsilon_i\dot{\varepsilon}_i^2}{(1 + \varepsilon_i^2)^2} \quad (21)$$

Remark 1. The value of $\varepsilon_i(t)$ will be big if $\psi(\varepsilon_i)$ deviates somewhat from the origin because the tan-type function (19) is sensitive to $\psi(\varepsilon_i)$. Our suggested controller, in conjunction with the tan-type error transformation function, guarantees that the error is kept within a certain, limited radius surrounding the origin.

In summary, our constraint control strategy is to make the tracking error converge to a predetermined region in a fixed time by means of the homogeneous embryo transformation, which facilitates the subsequent controller design and can be expressed as $\lim_{t \rightarrow T_i} |e_i(t) \leftrightarrow \varepsilon_i(t)| \leq \eta_i$.

Definition 2. Think about the system that follows [37],

$$\dot{x}(t) = f(x(t)), x(0) = 0, f(0) = 0 \quad (22)$$

where $x \in \mathbb{R}^n$, $f(x) : U_0 \rightarrow \mathbb{R}^n$ is continuous in the open neighborhood of the origin. Assume that the system (22) has a unique solution in forward time for all initial conditions.

For any real numbers $\alpha, \beta > 0, \rho > 1, 0 < q < 1$, there exists a positive definite Lyapunov function $V(x) = 0 \Rightarrow x = 0$, and there is $V(x) \leq -\alpha V'(x) - \beta V'(x)$. Then, the system (22) is globally fixed-time stable and the stability time function is bounded to be

$$T_i \leq T_{\max} = \frac{1}{\alpha(p-1)} + \frac{1}{\beta(q-1)} \quad (23)$$

Definition 3. For any real number $x_i (i = 1, 2, \dots, n)$, there exist real numbers $0 < q \leq 1$, such that the following inequality is true [38]:

$$(|x_1| + \dots + |x_n|)^q \leq |x_1|^q + \dots + |x_n|^q \quad (24)$$

where $q = \frac{q_1}{q_2}$ and both q_1 and q_2 are positive odd numbers.

Definition 4. For any real number $x_i (i = 1, 2, \dots, n)$, there exist real numbers $p > 1$, such that the following inequality holds [39]:

$$\sum_{i=1}^n |x_i|^p \leq \left(\sum_{i=1}^n |x_i| \right)^p \leq n^{p-1} \sum_{i=1}^n |x_i|^p \quad (25)$$

3.2. Neural Network Controller Design

Let $x_d(t)$ be the ideal trajectory in the workspace, $x(t)$ be the actual trajectory, and $x_r(t)$ be the reference trajectory after transformation. The error vector $\varepsilon(t)$ of the robot in the workspace is

$$\varepsilon(t) = x(t) - x_d(t) \quad (26)$$

$$\dot{x}_r(t) = \dot{x}_d(t) + \Lambda \varepsilon(t) \quad (27)$$

Define

$$r(t) = \dot{x}_r(t) - \dot{x}(t) = \dot{\varepsilon}(t) + \Lambda \varepsilon(t) \quad (28)$$

where the matrix Λ is positive definite.

From Section 2.2, the dynamic equation of the manipulator in the workspace after modeling with the neural network can be written as

$$H_x(\phi) \ddot{x} + C_x(\phi, \dot{\phi}) \dot{x} + g_x(\phi) = \tau_x \quad (29)$$

where

$$H_{xkj}(\phi) = \theta_{kj}^T \xi_{kj}(\phi) + \varepsilon_{hkj}(\phi) \quad (30)$$

$$C_{xkj}(\phi, \dot{\phi}) = n_{kj}^T \zeta_{kj}(\phi) + \varepsilon_{ckj}(\phi) \quad (31)$$

$$g_{xk}(\phi) = m_k^T \zeta_k(\phi) + \varepsilon_{gk}(\phi) \quad (32)$$

Using GL matrices and their multiplication operations [36], $H_x(\phi)$ can be written as

$$H_x(\phi) = \left[\{\Theta\}^T \cdot \{\Xi(\phi)\} \right] + E_h(\phi) \quad (33)$$

where $\{\Theta\}$ and $\Xi(\phi)$ are GL matrices whose elements are θ_{kj} and $\xi_{kj}(\phi)$, respectively, and $E_h(\phi) \in R^{n \times n}$ is the matrix of the modeling error $\epsilon_{hkj}(\phi)$.

Similarly, a GL matrix transformation of $C_x(\phi, \dot{\phi})$ and $g_{xk}(\phi)$ has

$$C_x(\phi, \dot{\phi}) = [\{N\}^T \cdot \{\Gamma(\Phi)\}] + E_c(\Phi) \quad (34)$$

$$g_x(\phi) = [\{M\}^T \cdot \{\Pi(\phi)\}] + E_g(\phi) \quad (35)$$

where $\{N\}$, $\Gamma(\Phi)$, $\{M\}$, and $\{\Pi(\phi)\}$ are GL matrices whose elements are n_{kj} , $\zeta_{kj}(\Phi)$, β_k , and $\zeta_k(\phi)$ in Equations (31) and (32), and $E_c(\Phi) \in R^{n \times n}$ and $E_g(\phi) \in R^{n \times n}$ are modeling error matrices.

Bringing Equations (33)–(35) into Equation (29), we obtain

$$[\{\Theta\}^T \cdot \{\Xi(\phi)\} + E_h(\phi)] \ddot{x} + [\{N\}^T \cdot \{\Gamma(\Phi)\} + E_c(\Phi)] \dot{x} + [\{M\}^T \cdot \{\Pi(\phi)\} + E_g(\phi)] = \tau_x \quad (36)$$

Selecting $(\hat{\cdot})$ to stand in for the anticipated value of $(\cdot)$, define $(\tilde{\cdot}) = (\cdot) - (\hat{\cdot})$. The design controller is

$$\tau_x = [\{\hat{\Theta}\}^T \cdot \{\Xi(\phi)\}] \ddot{x}_r + [\{\hat{N}\}^T \cdot \{\Gamma(\Phi)\}] \dot{x}_r + [\{\hat{M}\}^T \cdot \{\Pi(\phi)\}] + K_r + E_s \text{sgn}(r) \quad (37)$$

where $K_r \in R^{n \times n} > 0$ and $E_s > \|E\| = E_h(\phi) \ddot{x}_r + E_c(\Phi) \dot{x}_r + E_g(\phi)$. Model-based control is represented by the first three terms of the controller, and proportional differential (PD) control is represented by the term K_r , and the robust term that suppresses the neural network's modeling error is the last term of the control rule.

Combining Equations (36) and (37), it is obtained that

$$\begin{aligned} & [\{\Theta\}^T \cdot \{\Xi(\phi)\} + E_h(\phi)] \ddot{x} + [\{N\}^T \cdot \{\Gamma(\Phi)\} + E_c(\Phi)] \dot{x} + [\{M\}^T \cdot \{\Pi(\phi)\} + E_g(\phi)] \\ &= [\{\hat{\Theta}\}^T \cdot \{\Xi(\phi)\}] \ddot{x}_r + [\{\hat{N}\}^T \cdot \{\Gamma(\Phi)\}] \dot{x}_r + [\{\hat{M}\}^T \cdot \{\Pi(\phi)\}] + K_r + E_s \text{sgn}(r) \end{aligned} \quad (38)$$

Bringing $\dot{x} = \dot{x}_r - r$ and $\ddot{x} = \ddot{x}_r - \dot{r}$ into the above equation yields

$$\begin{aligned} & [\{\Theta\}^T \cdot \{\Xi(\phi)\} + E_h(\phi)] (\ddot{x}_r - \dot{r}) + [\{N\}^T \cdot \{\Gamma(\Phi)\} + E_c(\Phi)] (\dot{x}_r - r) + [\{M\}^T \cdot \{\Pi(\phi)\} + E_g(\phi)] \\ &= [\{\hat{\Theta}\}^T \cdot \{\Xi(\phi)\}] \ddot{x}_r + [\{\hat{N}\}^T \cdot \{\Gamma(\Phi)\}] \dot{x}_r + [\{\hat{M}\}^T \cdot \{\Pi(\phi)\}] + K_r + E_s \text{sgn}(r) \end{aligned} \quad (39)$$

Simplification of the above equation yields

$$\begin{aligned} & [\{\Theta\}^T \cdot \{\Xi(\phi)\} + E_h(\phi)] \dot{r} + [\{N\}^T \cdot \{\Gamma(\Phi)\} + E_c(\Phi)] r + K_r + E_s \text{sgn}(r) \\ &= [\{\tilde{\Theta}\}^T \cdot \{\Xi(\phi)\}] \ddot{x}_r + [\{\tilde{N}\}^T \cdot \{\Gamma(\Phi)\}] \dot{x}_r + [\{\tilde{M}\}^T \cdot \{\Pi(\phi)\}] + E \end{aligned} \quad (40)$$

Substituting Equations (33) and (34), we obtain

$$\begin{aligned} & H_x(\phi) \dot{r} + C_x(\phi, \dot{\phi}) r + K_r + E_s \text{sgn}(r) \\ &= [\{\tilde{\Theta}\}^T \cdot \{\Xi(\phi)\}] \ddot{x}_r + [\{\tilde{N}\}^T \cdot \{\Gamma(\Phi)\}] \dot{x}_r + [\{\tilde{M}\}^T \cdot \{\Pi(\phi)\}] + E \end{aligned} \quad (41)$$

The stability of the closed-loop system (41) is provided by the following theorem.

Theorem 1. For a closed-loop system Equation (41) if $K_r > 0$, $E_s > \|E\|$ and the adaptive law satisfies

$$\dot{\hat{\theta}}_k = A_k \cdot \{\xi_k(\phi)\} \ddot{x}_r r_k \quad (42)$$

$$\dot{\hat{n}}_k = B_k \cdot \{\zeta_k(\Phi)\} \dot{x}_r r_k \quad (43)$$

$$\dot{\hat{m}}_k = C_k \cdot \{\zeta_k(\phi)\} r_k \quad (44)$$

where $A_k = A_k^T > 0$; $B_k = B_k^T$; $C_k = C_k^T$, and $\hat{\theta}_k$, $\hat{n}_k$, $\hat{m}_k$ are vectors of $\hat{\theta}_{kj}$, $\hat{n}_{kj}$, and $\hat{m}_{kj}$, respectively, then $\hat{\theta}_k$, $\hat{n}_k$, $\hat{m}_k \in L_\infty$, $\varepsilon \in L_2^n \cap L_\infty^n$, and ε is continuous, and $\varepsilon, \dot{\varepsilon} \rightarrow 0$ when $t \rightarrow \infty$.

Proof of Theorem 1. Consider the following Lyapunov function

$$V = \frac{1}{2} r^T H_x(\phi) r + \frac{1}{2} \sum_{k=1}^n \tilde{\theta}_k^T A_k^{-1} \tilde{\theta}_k + \frac{1}{2} \sum_{k=1}^n \tilde{n}_k^T B_k^{-1} \tilde{n}_k + \frac{1}{2} \sum_{k=1}^n \tilde{m}_k^T C_k^{-1} \tilde{m}_k \quad (45)$$

where A_k, B_k, C_k are positive definite matrices. Derivation of Equation (45) yields

$$\dot{V} = r^T H_x(\phi) \dot{r} + \frac{1}{2} r^T \dot{H}_x(\phi) r + \sum_{k=1}^n \tilde{\theta}_k^T A_k^{-1} \dot{\tilde{\theta}}_k + \sum_{k=1}^n \tilde{n}_k^T B_k^{-1} \dot{\tilde{n}}_k + \sum_{k=1}^n \tilde{m}_k^T C_k^{-1} \dot{\tilde{m}}_k \quad (46)$$

Since the matrix $\dot{H}_x(\phi) - 2C_x(\phi, \dot{\phi})$ is skew-symmetric, then $r^T (\dot{H}_x(\phi) - 2C_x(\phi, \dot{\phi})) r = 0$.

Using the fact that $\dot{\tilde{\theta}}_k = -\dot{\hat{\theta}}_k$, $\dot{\tilde{n}}_k = -\dot{\hat{n}}_k$ and $\dot{\tilde{m}}_k = -\dot{\hat{m}}_k$, substituting into the above equation

$$\dot{V} = r^T (H_x(\phi) \dot{r} + C_x(\Phi) r) - \sum_{k=1}^n \tilde{\theta}_k^T A_k^{-1} \dot{\hat{\theta}}_k - \sum_{k=1}^n \tilde{n}_k^T B_k^{-1} \dot{\hat{n}}_k - \sum_{k=1}^n \tilde{m}_k^T C_k^{-1} \dot{\hat{m}}_k \quad (47)$$

Substituting (41) yields

$$\begin{aligned} \dot{V} = & -r^T K_r - E_s r^T \text{sgn}(r) + \sum_{k=1}^n \left\{ \tilde{\theta}_k^T \right\} \cdot \left\{ \zeta_k(\phi) \right\} \ddot{x}_r r_k \\ & + \sum_{k=1}^n \left\{ \tilde{n}_k^T \right\} \cdot \left\{ \zeta_k(\Phi) \right\} \dot{x}_r r_k + \sum_{k=1}^n \left\{ \tilde{m}_k^T \right\} \cdot \left\{ \zeta_k(\phi) \right\} r_k \\ & + r^T E - \sum_{k=1}^n \tilde{\theta}_k^T A_k^{-1} \dot{\hat{\theta}}_k - \sum_{k=1}^n \tilde{n}_k^T B_k^{-1} \dot{\hat{n}}_k - \sum_{k=1}^n \tilde{m}_k^T C_k^{-1} \dot{\hat{m}}_k \end{aligned} \quad (48)$$

Substituting the adaptive laws (42), (43), and (44) into the above equation and combining them with $E_s > \|E\|$, we obtain

$$\dot{V} = -r^T K_r - E_s r^T \text{sgn}(r) + r^T E \leq 0 \quad (49)$$

Since $V \geq 0$, $\dot{V} \leq 0$, hence, r , $\tilde{\theta}_k$, $\tilde{n}_k$, and $\tilde{m}_k$ are bounded. When $\dot{V} \equiv 0$ is satisfied $r = 0$; the closed-loop system is asymptotically stable according to the LaSalle invariance principle. When $t \rightarrow \infty$, $r \rightarrow 0$, there is $\varepsilon, \dot{\varepsilon} \rightarrow 0$. $\square$

4. Simulation Experiment Verification

To illustrate the effectiveness of the validation algorithm, a planar two-link manipulator, widely used in the literature, is considered in our simulation study. The kinetic equations of the manipulator are

$$H_x(\phi) \ddot{x} + C_x(\phi, \dot{\phi}) \dot{x} + g_x(\phi) = \tau_x \quad (50)$$

where

$$H_x(\phi) = J^{-T}(\phi) H(\phi) J^{-1}(\phi) \quad (51)$$

$$g_x(\phi) = J^{-T}(\phi) g(\phi) \quad (52)$$

$$C_x(\phi, \dot{\phi}) = J^{-T}(\phi) (C(\phi, \dot{\phi}) - H(\phi) J^{-1}(\phi) \dot{J}(\phi)) J^{-1}(\phi) \quad (53)$$

$$H(\phi) = \begin{bmatrix} p_1 + p_2 + 2p_3 \cos \phi_2 & p_2 + p_3 \cos \phi_2 \\ p_2 + p_3 \cos \phi_2 & p_2 \end{bmatrix} \quad (54)$$

$$C(\phi, \dot{\phi}) = \begin{bmatrix} -p_3 \dot{\phi}_2 \sin \phi_2 & -p_3 (\dot{\phi}_1 + \dot{\phi}_2) \sin \phi_2 \\ p_3 \dot{\phi}_1 \sin \phi_2 & 0 \end{bmatrix} \quad (55)$$

$$g(\phi) = \begin{bmatrix} p_4 g \cos \phi_1 + p_5 g \cos(\phi_1 + \phi_2) \\ p_5 g \cos(\phi_1 + \phi_2) \end{bmatrix} \quad (56)$$

$$J(\phi) = \begin{bmatrix} -l_1 \sin(\phi_1) - l_2 \sin(\phi_1 + \phi_2) & -l_2 \sin(\phi_1 + \phi_2) \\ l_1 \cos(\phi_1) + l_2 \cos(\phi_1 + \phi_2) & l_2 \cos(\phi_1 + \phi_2) \end{bmatrix} \quad (57)$$

$$\begin{aligned} \dot{J}(\phi) = & \begin{bmatrix} -l_1 \cos(\phi_1) - l_2 \cos(\phi_1 + \phi_2) & -l_2 \cos(\phi_1 + \phi_2) \\ -l_1 \sin(\phi_1) - l_2 \sin(\phi_1 + \phi_2) & -l_2 \sin(\phi_1 + \phi_2) \end{bmatrix} \dot{\phi}_1 \\ & + \begin{bmatrix} -l_2 \cos(\phi_1 + \phi_2) & -l_2 \cos(\phi_1 + \phi_2) \\ -l_2 \sin(\phi_1 + \phi_2) & -l_2 \sin(\phi_1 + \phi_2) \end{bmatrix} \dot{\phi}_2 \end{aligned} \quad (58)$$

Take $P = [p_1 \ p_2 \ p_3 \ p_4 \ p_5] = [2.90 \ 0.76 \ 0.87 \ 3.04 \ 0.87]^T$. Since the tracking trajectory is a right-angle coordinate in the workspace, not an angular position in the joint space, the dijoint angular position should be converted to the joint end Cartesian coordinate in the workspace according to Equations (4) and (5). The Jacobian matrix $J(\phi)$ is shown in Equations (57) and (58).

The ideal tracking trajectory in the workspace is taken as $x_{d1}(t) = 0.5\cos(0.5 * \pi t) + 1.0$, $x_{d2}(t) = 0.5\sin(0.5 * \pi t) + 1.0$, which is a circle with radius 0.5 m and a center at $(x_1, x_2) = (1.0, 1.0)$ m. For the initial condition, the end-effector of the manipulator is at the center of the circle, i.e., $x(0) = [1.0 \ 1.0]$ m, $\dot{x}(0) = [0 \ 0]$ m/s.

For the approximation of each element of $H_x(\phi)$, $g_x(\phi)$, and $C_x(\phi, \dot{\phi})$, the RBF neural network accepts ϕ and $\dot{\phi}$ as input, and its hidden layer has seven nodes by design.

All Gaussian functions' parameters are assumed to be $b_i = 10$ and $c_i = [-1.5 \ -1.0 \ -0.5 \ 0 \ 0.5 \ 1.0 \ 1.5]$, putting the neural network's starting value at 0. Equations (37) and (42)–(44) employ the adaptive law and the control law, respectively. The controller gains are chosen as $K = \begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix}$ and $E_s = 0.5$. By Lemma 2,

$\Lambda = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}$ is taken and the parameters of the adaptive law are taken as $A_k = \text{diag}\{2.0\}$, $B_k = \text{diag}\{0.10\}$ and $C_k = \text{diag}\{5.0\}$.

The simulation results are shown in Figures 3–11. Figure 3 illustrates the trajectory tracking performance of the proposed algorithm applied to the manipulator and compares it with the PID and robust control algorithms under the same conditions. The results indicate that all three control methods can achieve the desired trajectory tracking. However, the proposed workspace manipulator adaptive neural network control method based on time preset performance (hereafter referred to as FTTP-NN) demonstrates faster tracking speed and higher accuracy, achieving precise tracking within the preset 1 s. Notably, the tracking errors of the PID and robust control methods exceed the boundaries during the initial phase, whereas the FTTP-NN method ensures that the tracking error remains constrained within the boundaries at all times, thereby improving both the transient and steady-state performance of the system.

Figures 4 and 5, respectively, show the errors in the x_1 and x_2 directions under the Cartesian coordinate system of the joint ends in the workspace. The FTTP-NN method demonstrates superior performance compared to both PID and robust control methods. The tracking errors in the x_1 and x_2 directions are significantly reduced, and the FTTP-NN method achieves stable and accurate tracking within the specified time frame.

Figure 6 shows the ideal and estimated values of the kinetic parameters H_x , C_x , and g_x over time. The estimation of the adaptive neural network for the kinetic parameters in the FTTP-NN algorithm maintains a high consistency with the ideal values, especially

in the steady-state part of the system response. This demonstrates the robustness of the FTTP-NN method in accurately estimating and adapting to the system's kinetic parameters. In terms of velocity tracking, Figure 7 demonstrates the velocity tracking capabilities of the three methods. When the system reaches a steady state, the velocity profiles show precise tracking of the desired velocity with minimal steady-state error. Compared to the PID and robust control algorithms, the FTTP-NN control strategy significantly reduces the steady-state error, demonstrating its superior performance in maintaining accurate velocity tracking. The joint torque analysis in Figure 8 shows that the FTTP-NN method can quickly reduce the high torque peaks caused by initial conditions and maintain a stable torque output thereafter.

The technique presented in this work may meet the criterion of trajectory tracking accuracy after homozygous error conversion, as shown in Figure 9. After appropriate transformation, the position tracking error ε_i is confined within the preset limits, offering explicit guidance for designing and tuning the control system. Error analysis in Figures 10 and 11 shows that adjusting the convergence boundary η_i and the time parameter T_i enables the FTTP-NN method to maintain rapid and precise tracking control. The impact of various parameters on the dynamics of position tracking error is further elucidated, highlighting the FTTP-NN controller's adaptability to diverse control tasks.

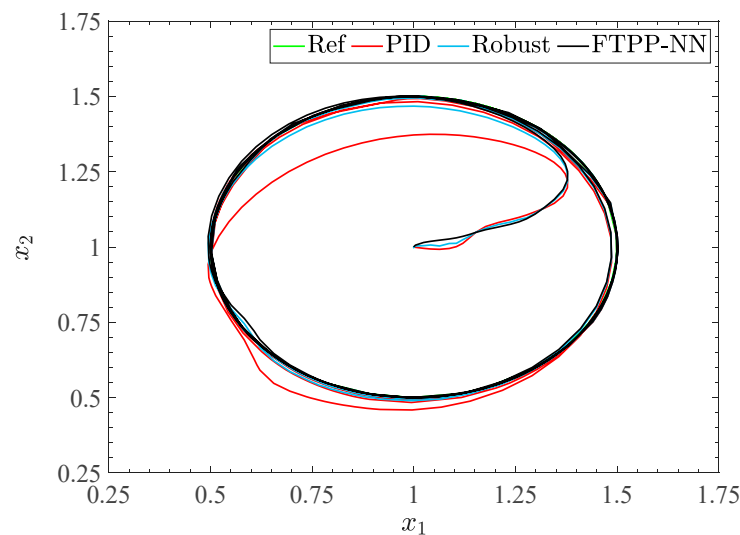


Figure 3. Tracking the trajectory of the end joint.

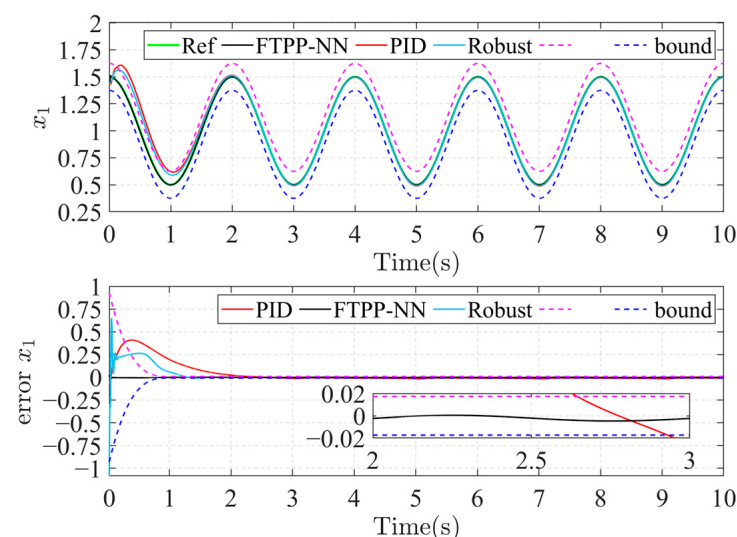


Figure 4. Trajectories and errors in the x_1 direction.

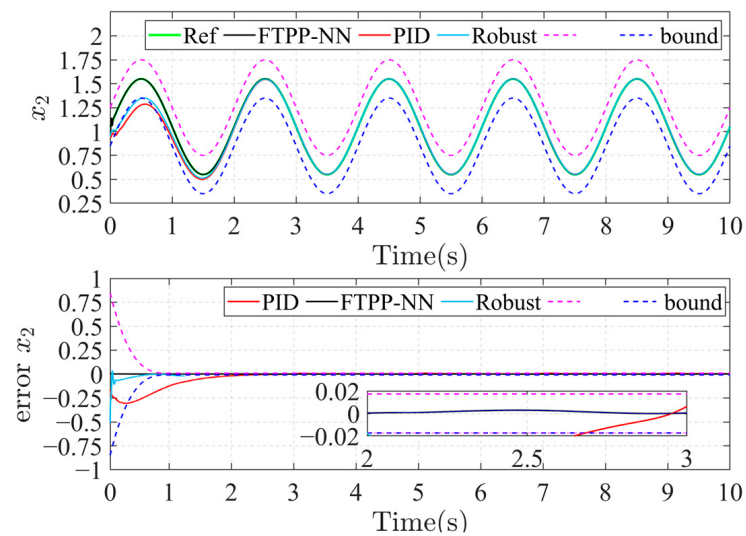


Figure 5. Trajectories and errors in the x_2 direction.

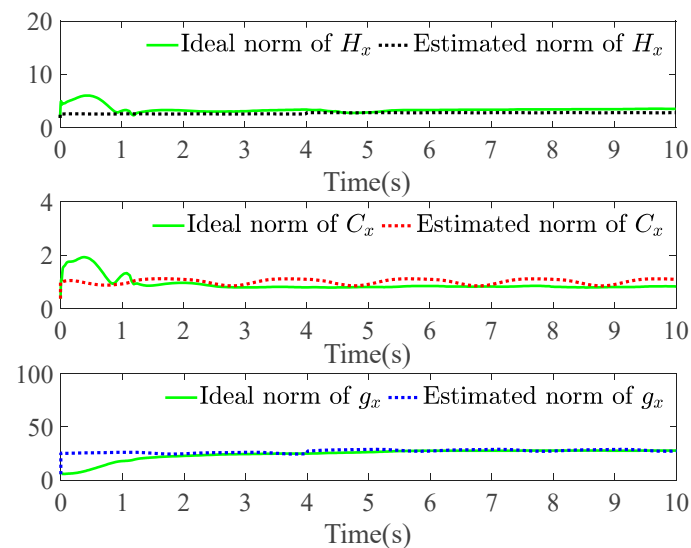


Figure 6. Approximation of kinetic parameters for H_x , C_x , and g_x .

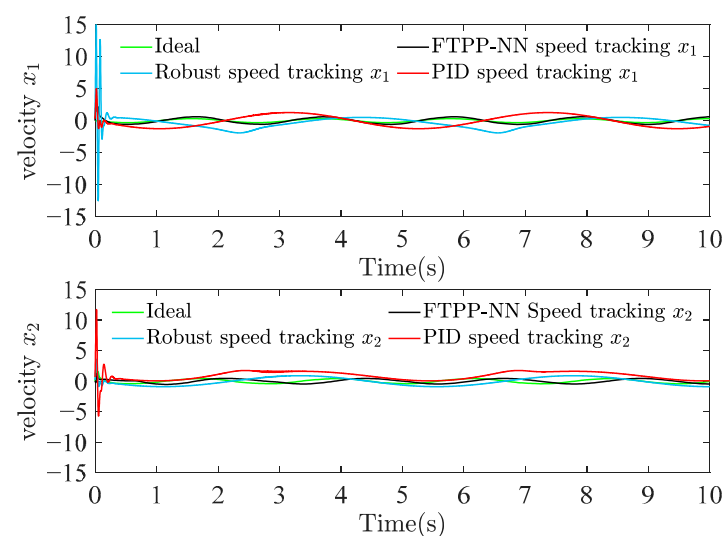


Figure 7. Angular velocity tracking of the end joints.

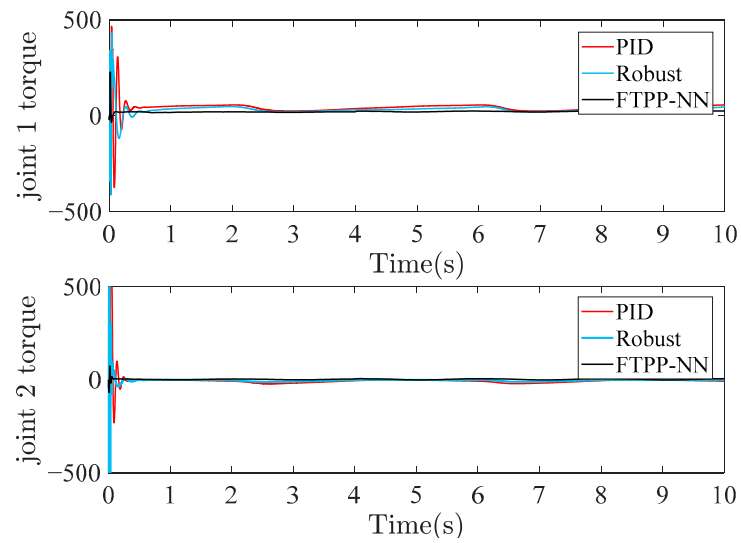


Figure 8. Control inputs for Joint 1 and Joint 2.

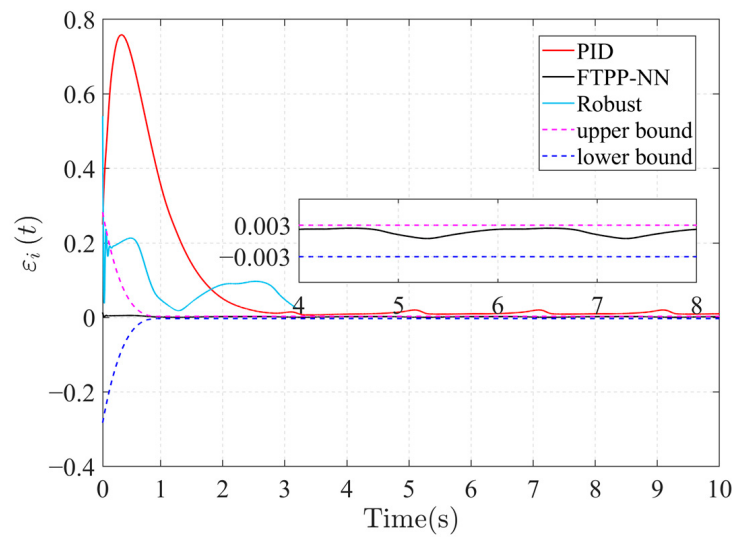


Figure 9. Trajectory tracking error for different control algorithms.

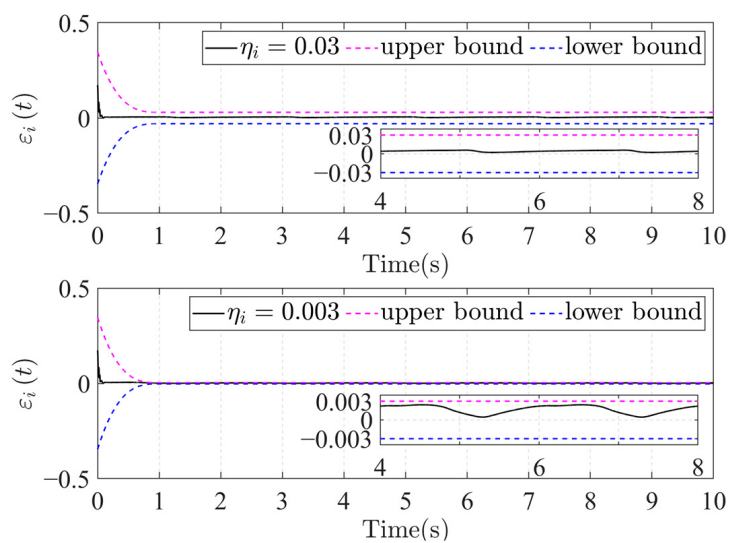


Figure 10. Trajectory tracking error for different preset boundaries.

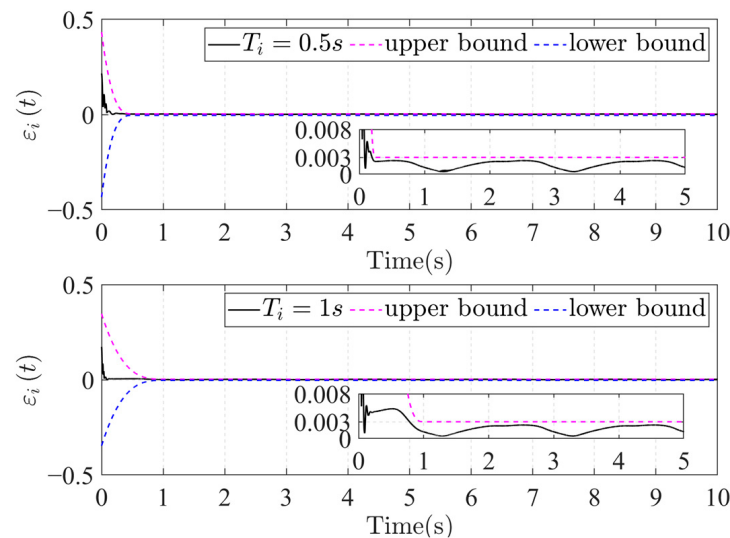


Figure 11. Trajectory tracking error for different preset times.

5. Conclusions

This paper introduces a fixed-time, performance-constrained control algorithm using an RBF neural network for workspace manipulators, aimed at enhancing the convergence speed and accuracy for trajectory tracking within defined time and steady-state limits. The controller design features a tangent-type error transformation through homogeneous embryonic transformation, directing the tracking error towards a specified region. Additionally, the control strategy is woven into a fixed-time stability framework, ensuring all system states are confined within a bounded region in a finite timeframe. Lastly, dynamic parameters estimated through a neural network are fed back to the controller for enhanced parameter optimization. The error from the fixed-time performance control strategy is integrated with neural network control in designing the controller, securing swift and precise manipulator trajectory tracking. Simulation results confirm the effectiveness and superior performance of the proposed controller. However, the current study does not include experimental validation, which is a limitation. In practice, akin to widely used multi-degree-of-freedom industrial manipulators, the next step involves extending this paper's control method to such systems and conducting experimental studies to validate the simulation results. Additionally, we plan to explore time-specific performance-constrained trajectory tracking for a broader range of nonlinear mechanical systems.

Author Contributions: Conceptualization, X.C. and H.Z.; methodology, X.C. and S.Z.; validation, S.Z. and J.Z.; investigation, X.L.; writing—original draft preparation, X.C.; writing—review and editing, X.C. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Scientific Research Startup Fund of West Anhui University (WGKQ2020010005).

Data Availability Statement: Data are contained within the article.

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

References

1. Chen, Z.; Zhou, S.; Shen, C.; Lyu, L.; Zhang, J.; Yao, B. Observer-Based Adaptive Robust Precision Motion Control of a Multi-Joint Hydraulic Manipulator. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 1213–1226. [\[CrossRef\]](#)
2. Li, Z.; Li, S. Recursive Recurrent Neural Network: A Novel Model for Manipulator Control with Different Levels of Physical Constraints. *CAAI Trans. Intell. Technol.* **2023**, *8*, 622–634. [\[CrossRef\]](#)
3. Ahmed, S.; Azar, A.T. Adaptive Fractional Tracking Control of Robotic Manipulator Using Fixed-Time Method. *Complex Intell. Syst.* **2024**, *10*, 369–382. [\[CrossRef\]](#)
4. Lee, J.; Chang, P.H.; Yu, B.; Jin, M. An Adaptive PID Control for Robot Manipulators Under Substantial Payload Variations. *IEEE Access* **2020**, *8*, 162261–162270. [\[CrossRef\]](#)

5. Loucif, F.; Kechida, S.; Sebbagh, A. Whale Optimizer Algorithm to Tune PID Controller for the Trajectory Tracking Control of Robot Manipulator. *J. Braz. Soc. Mech. Sci. Eng.* **2020**, *42*, 1. [[CrossRef](#)]
6. Chotikunnan, P.; Chotikunnan, R. Dual Design PID Controller for Robotic Manipulator Application. *J. Robot. Control* **2023**, *4*, 23–34. [[CrossRef](#)]
7. Ahmed, A.S.; Kadhim, S.K. A Comparative Study Between Convolution and Optimal Backstepping Controller for Single Arm Pneumatic Artificial Muscles. *J. Robot. Control* **2022**, *3*, 769–778. [[CrossRef](#)]
8. Pan, Y.; Wang, H.; Li, X.; Yu, H. Adaptive Command-Filtered Backstepping Control of Robot Arms with Compliant Actuators. *IEEE Trans. Control Syst. Technol.* **2017**, *26*, 1149–1156. [[CrossRef](#)]
9. Baek, J.; Jin, M.; Han, S. A New Adaptive Sliding-Mode Control Scheme for Application to Robot Manipulators. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3628–3637. [[CrossRef](#)]
10. Ripamonti, F.; Orsini, L.; Resta, F. A Nonlinear Sliding Surface in Sliding Mode Control to Reduce Vibrations of a Three-Link Flexible Manipulator. *J. Vib. Acoust.* **2017**, *139*, 051005. [[CrossRef](#)]
11. Pezzato, C.; Ferrari, R.; Corbato, C.H. A Novel Adaptive Controller for Robot Manipulators Based on Active Inference. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2973–2980. [[CrossRef](#)]
12. Huang, H.; Yang, C.; Chen, C.L.P. Optimal Robot–Environment Interaction Under Broad Fuzzy Neural Adaptive Control. *IEEE Trans. Cybern.* **2020**, *51*, 3824–3835. [[CrossRef](#)]
13. Tang, Z.Q.; Heung, H.L.; Tong, K.Y.; Li, Z. Model-Based Online Learning and Adaptive Control for a “Human-Wearable Soft Robot” Integrated System. *Int. J. Robot. Res.* **2021**, *40*, 256–276. [[CrossRef](#)]
14. Calinon, S. Gaussians on Riemannian Manifolds: Applications for Robot Learning and Adaptive Control. *IEEE Robot. Autom. Mag.* **2020**, *27*, 33–45. [[CrossRef](#)]
15. Nubert, J.; Köhler, J.; Berenz, V.; Allgöwer, F.; Trimpe, S. Safe and Fast Tracking on a Robot Manipulator: Robust MPC and Neural Network Control. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3050–3057. [[CrossRef](#)]
16. Emmanuel Agbaraji, C.; Henrietta Udeani, U.; Chibueze Inyama, H.; Chikodi Okezie, C. Robust Control for a 3DOF Articulated Robotic Manipulator Joint Torque Under Uncertainties. *J. Eng. Res. Rep.* **2020**, *9*, 1–3. [[CrossRef](#)]
17. Van, M.; Ceglarek, D. Robust Fault Tolerant Control of Robot Manipulators with Global Fixed-Time Convergence. *J. Frankl. Inst.* **2021**, *358*, 699–722. [[CrossRef](#)]
18. Dai, L.; Yu, Y.; Zhai, D.H.; Huang, T.; Xia, Y. Robust Model Predictive Tracking Control for Robot Manipulators with Disturbances. *IEEE Trans. Ind. Electron.* **2020**, *68*, 4288–4297. [[CrossRef](#)]
19. 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.* **2020**, *29*, 2388–2398. [[CrossRef](#)]
20. Wang, Z.; Zou, L.; Su, X.; Luo, G.; Li, R.; Huang, Y. Hybrid Force/Position Control in Workspace of Robotic Manipulator in Uncertain Environments Based on Adaptive Fuzzy Control. *Robot. Auton. Syst.* **2021**, *145*, 103870. [[CrossRef](#)]
21. Sun, Y.; Chen, M.; Gao, C.; Wu, L. Output Feedback Command Filtered Fuzzy Controller Design for Uncertain Strict-Feedback Nonlinear Systems with Unmodeled Dynamics and Event-Triggered Strategy. *Eng. Res. Express* **2024**, *6*, 025315. [[CrossRef](#)]
22. Urrea, C.; Kern, J.; Alvarado, J. Design and Evaluation of a New Fuzzy Control Algorithm Applied to a Manipulator Robot. *Appl. Sci.* **2020**, *10*, 7482. [[CrossRef](#)]
23. Ha, S.; Liu, H.; Li, S.; Liu, A. Backstepping-Based Adaptive Fuzzy Synchronization Control for a Class of Fractional-Order Chaotic Systems with Input Saturation. *Int. J. Fuzzy Syst.* **2019**, *21*, 1571–1584. [[CrossRef](#)]
24. Xu, F.; Tang, L.; Liu, Y.J. Tangent Barrier Lyapunov Function-Based Constrained Control of Flexible Manipulator System with Actuator Failure. *Int. J. Robust Nonlinear Control* **2021**, *31*, 8523–8536. [[CrossRef](#)]
25. Yin, X.; Pan, L. Direct Adaptive Robust Tracking Control for 6 DOF Industrial Robot with Enhanced Accuracy. *ISA Trans.* **2018**, *72*, 178–184. [[CrossRef](#)] [[PubMed](#)]
26. Gao, S.; Ning, B.; Dong, H. Adaptive Neural Control with Intercepted Adaptation for Time-Delay Saturated Nonlinear Systems. *Neural Comput. Appl.* **2015**, *26*, 1849–1857. [[CrossRef](#)]
27. Yu, C.; Xu, L.; Sun, H. Research on Manipulator Motion Control System Based on Preset Performance Control Algorithm. *Int. Conf. Robot. Intell. Syst.* **2020**, *20*, 28–32.
28. Liu, G.; Sun, N.; Yang, T.; Fang, Y. Reinforcement Learning-Based Prescribed Performance Motion Control of Pneumatic Muscle Actuated Robotic Arms with Measurement Noises. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *53*, 1801–1812. [[CrossRef](#)]
29. Ma, H.; Zhou, Q.; Li, H.; Lu, R. Adaptive Prescribed Performance Control of a Flexible-Joint Robotic Manipulator with Dynamic Uncertainties. *IEEE Trans. Cybern.* **2021**, *52*, 12905–12915. [[CrossRef](#)]
30. Gao, M.; Ding, L.; Jin, X. ELM-Based Adaptive Faster Fixed-Time Control of Robotic Manipulator Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 4646–4658. [[CrossRef](#)]
31. Zhang, L.; Wang, Y.; Hou, Y.; Li, H. Fixed-Time Sliding Mode Control for Uncertain Robot Manipulators. *IEEE Access* **2019**, *7*, 149750–149763. [[CrossRef](#)]
32. Vo, A.T.; Truong, T.N.; Kang, H.J. An Adaptive Prescribed Performance Tracking Motion Control Methodology for Robotic Manipulators with Global Finite-Time Stability. *Sensors* **2022**, *22*, 7834. [[CrossRef](#)] [[PubMed](#)]
33. Li, X.H.; Hu, L.Y. Prescribed Performance Finite-Time H_∞ Tracking Control for a Class of p-Normal Form Nonlinear Systems. *Acta Autom. Sin.* **2021**, *47*, 2870–2880.

34. Xu, Z.; Li, S.; Zhou, X.; Zhou, S.; Cheng, T.; Guan, Y. Dynamic Neural Networks for Motion-Force Control of Redundant Manipulators: An Optimization Perspective. *IEEE Trans. Ind. Electron.* **2020**, *68*, 1525–1536. [[CrossRef](#)]
35. Shang, D.; Li, X.; Yin, M.; Li, F. Dynamic Modeling and Neural Network Compensation for Dual-Flexible Servo System with an Underactuated Hand. *J. Frankl. Inst.* **2023**, *360*, 10127–10164. [[CrossRef](#)]
36. Zhang, J.; Liu, W.; Li, L.; Li, Z. The Master Adaptive Impedance Control and Slave Adaptive Neural Network Control in Underwater Manipulator uncertainty teleoperation. *Ocean Eng.* **2018**, *165*, 465–479. [[CrossRef](#)]
37. Zuo, Z.; Tian, B.; Defoort, M.; Ding, Z. Fixed-Time Consensus Tracking for Multiagent Systems with High-Order Integrator Dynamics. *IEEE Trans. Autom. Control* **2018**, *63*, 563–570. [[CrossRef](#)]
38. Zheng, Z.; Xia, Y.; Fu, M. Attitude Stabilization of Rigid Spacecraft with Finite-Time Convergence. *Int. J. Robust Nonlinear Control* **2015**, *21*, 686–702.
39. Jin, X. Adaptive Fixed-Time Control for MIMO Nonlinear Systems with Asymmetric Output Constraints Using Universal Barrier Functions. *IEEE Trans. Autom. Control* **2019**, *64*, 3046–3053. [[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.