

Article

A Model-Free-Based Control Method for Robot Manipulators: Achieving Prescribed Performance and Ensuring Fixed Time Stability

Anh Tuan Vo , Thanh Nguyen Truong  and Hee-Jun Kang * 

Department of Electrical, Electronic and Computer Engineering, University of Ulsan,
Ulsan 44610, Republic of Korea; voanhtuan@mail.ulsan.ac.kr (A.T.V.);
truongthannguyen@mail.ulsan.ac.kr (T.N.T.)

* Correspondence: hjkang@ulsan.ac.kr

Abstract: This paper addresses three significant challenges in controlling robot manipulators: improving response time, minimizing steady-state errors and chattering, and enhancing controller robustness. It also focuses on eliminating the need for computing the robot's dynamic model and unknown functions, as well as achieving global fixed-time convergence and the prescribed performance for the control system. To achieve these objectives, a fixed-time sliding mode function is designed, which uses transformation errors to achieve prescribed control performance, with adjustments made to the maximum overshoot, convergence time, and tracking errors to keep them within predefined bounds. Additionally, a radial basis function neural network (RBFNN) is used to eliminate the need for knowledge of the robot's dynamical properties and uncertain terms, which also reduces negative chattering. Finally, a novel fixed-time terminal sliding mode control (TSMC) algorithm is developed for robot manipulators without using their dynamical model. The fixed-time stability of the control system is thoroughly demonstrated by applying Lyapunov criteria and conducting simulations on a robot manipulator to showcase its effectiveness.

Keywords: control of robots; neural networks; fixed-time control; prescribed performance control



Citation: Vo, A.T.; Truong, T.N.; Kang, H.-J. A Model-Free-Based Control Method for Robot Manipulators: Achieving Prescribed Performance and Ensuring Fixed Time Stability. *Appl. Sci.* **2023**, *13*, 8939. <https://doi.org/10.3390/app13158939>

Academic Editor: Mohd Rizal Arshad

Received: 13 June 2023

Revised: 28 July 2023

Accepted: 2 August 2023

Published: 3 August 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

Robotic manipulators have become indispensable in the manufacturing industry, revolutionizing productivity and efficiency across a wide range of applications. These versatile machines play crucial roles in diverse manufacturing processes, including assembly, packaging, painting, material handling, and quality inspection. They find widespread use in industries such as automotive, electronics, pharmaceuticals, food processing, and logistics, where they excel at performing tasks like precision welding, cutting, drilling, palletizing, and order fulfillment [1–4]. As the field of robotics continues to expand its applications, it faces challenges in operating within unstructured environments characterized by nonlinear dynamics and influenced by environmental factors such as gravity, friction, and external disturbances. Overcoming these challenges is essential to ensure the reliable and optimal performance of manipulators in such complex scenarios. This has prompted researchers in academia and industry to develop enhanced control methods that improve performance and enhance the stability of robot systems under negative influences. These advancements aim to ensure the robust and secure operation of robotic manipulators, safeguarding their efficiency and effectiveness in manufacturing settings.

The robustness and reliability of robots can be enhanced through various control methodologies, such as PID control [5], computed torque control (CTC) [6,7], fault-tolerant control (FTC) with PID [8,9], neural network (NN)-based control [10,11], disturbance observer-based control [12–14], sliding mode control (SMC) [15–17], and different forms of the SMC, like higher-order SMC [18,19] and integral sliding mode control (ISMC) [20–22].

Among them, SMC and its various versions exhibit exceptional characteristics, including robustness against disturbances and uncertainties, finite-time convergence, and simplicity in design. These control techniques have found extensive applications across diverse systems, ranging from robot manipulators and underwater vehicles to spacecraft and electrical motors. However, ensuring fixed-time convergence for the system using conventional SMC methods is generally challenging. In addition, one primary hurdle is the occurrence of chattering, characterized by unwanted high-frequency oscillations or switching in the control signal due to the control law's discontinuity. Chattering can adversely impact system performance and result in mechanical wear.

To address these limitations and ensure that the controlled error variables converge to equilibrium in a predefined time period, fixed-time control approaches have been introduced for robots [23–26]. These controllers offer the advantage of achieving convergence within a fixed duration, thereby enhancing performance and reliability in robotic systems. By employing fixed-time control strategies, tracking errors can be effectively mitigated, enabling the system to meet stringent timing requirements in diverse applications. Fixed-time algorithms are specifically designed to ensure a bounded convergence time for the system, regardless of the initial state values. They have been developed based on various control frameworks, including SMC and backstepping control. For instance, fixed-time SMC algorithms have been proposed in specific studies [24,26,27], while fixed-time backstepping controls have been developed in other research papers [28–30]. These approaches provide valuable tools for achieving robust and time-constrained control in robotic systems. To further improve tracking performance and overcome the limitations of fixed-time controllers, the development of fixed-time adaptive controllers has gained attention [31–33]. These innovative approaches offer a reliable and practical solution for controlling robot manipulators in real-world applications. Notably, fixed-time adaptive SMC was successfully applied to robot manipulators, as demonstrated in Ref. [34]. Additionally, fixed-time adaptive fault-tolerant control (FTC) techniques were developed specifically for robot manipulators, as highlighted in Ref. [31]. These adaptive control strategies not only ensure fixed-time convergence but also adaptively adjust control parameters to enhance the system's robustness and performance. Through these advancements, robot manipulators can achieve precise and efficient control even in challenging and dynamic environments.

Designing a control system for robots poses implementation challenges, as it typically requires knowledge of the robot's dynamic model or at least an upper-bound estimation of the undefined dynamics. To address these challenges, model-free controllers based on learning techniques have been proposed, such as neural networks (NNs) [13,35,36] or fuzzy logic systems (FLSs) [31], where learning methods have been widely used in control design to approximate unknown components in robot dynamics. A distributed control using adaptive fuzzy control for autonomous underwater vehicles (AUVs) was developed where FLS is used to approximate the unknown dynamics [29]. For achieving fixed-time stability in adaptive NN-based methods, several solutions have been proposed for robots [37–39]. Unfortunately, these studies still require the calculation of a part of the robotic system's dynamics. Additionally, in an attempt to achieve overall fixed-time stability, references [38,39] utilize the conversion constant of the NN instead of estimating its weight. This approach makes it resemble a normal adaptive controller, thereby diminishing the strong properties typically associated with NNs.

The prescribed performance control algorithm (PPCA) is a powerful control system design approach that involves specifying measurable outputs for a system and designing a controller to achieve the desired performance [13,40,41]. This approach is particularly useful in complex or uncertain systems where traditional control techniques may fall short. By prescribing the desired performance directly, the PPCA can offer increased robustness to uncertainties and disturbances, resulting in better overall performance. However, it is important to recognize that the PPCA does have limitations. For example, the system's performance can be sensitive to changes in system dynamics or operating conditions, and the accuracy of the system model is crucial for achieving desired performance. In

addition, some studies have pointed out potential singularity issues with certain used ETFs [39,40,42,43]. Therefore, to maximize its capabilities, the PPCA should be combined with other control methods.

After a thorough analysis of the challenges in controlling robot manipulators, we developed a novel control system with global fixed-time stability for robots. The proposed approach utilizes a fixed-time adaptive NN controller designed within TSMC and the PPCA framework to achieve global convergence of the controller in a fixed time, following Lyapunov criteria. This paper contributes in several ways:

- Our approach uses two separate pre-specified performance functions (PPFs) to set bounds for tracking errors. One PPF limits the convergence rate and steady-state error, while the other addresses overshoot and the steady-state error. This enlarges the performance space at a steady state compared to traditional methods. Additionally, the symmetric steady-state error boundaries ensure the zero-tracking error when the transformed error is zero, simplifying the ETF design. Our ETF is also free from singularity problems;
- The use of transformation errors as NN inputs, which can lead to faster approximation and improved accuracy in learning unknown models;
- The proposed method can minimize the high chattering effect produced by other conventional controllers, while achieving fixed-time convergence and prescribed performance;
- Unlike conventional adaptive NNs that solely focus on achieving fixed-time convergence of controlled errors, the proposed controller takes a novel approach by attaining fixed-time convergence for the adaptive rules of the NNs. This distinctive feature guarantees global fixed-time convergence for both controlled errors and adaptive rules, distinguishing it from conventional approaches.

This paper consists of several sections that address different aspects of the research topic. Section 2 is dedicated to providing preliminaries that cover the dynamic model of manipulators and the concept of fixed-time convergence. In Section 3, a model-based TSMC is designed using PPCA, focusing on their combined effectiveness. In Section 4, we present the proposed design of a model-free TSMC for robots. The primary objective is to investigate the prescribed performance of this controller in achieving fixed-time convergence. Section 5 is dedicated to presenting the simulation results of the developed algorithm, focusing on the evaluation and analysis of its performance. In conclusion, Section 6 provides a summary of the findings, thus concluding the paper.

2. Preliminaries

To facilitate the presentation, the paper introduces the following notations:

$$\text{Sig}^{\tilde{\zeta}}(E) = |E|^{\tilde{\zeta}} \text{sign}(E), \text{ where } \text{sign}(E) \text{ is defined as follows: } \text{sign}(E) = \begin{cases} 1 & \text{if } E > 0 \\ 0 & \text{if } E = 0 \\ -1 & \text{otherwise} \end{cases}$$

with $\tilde{\zeta} > 0$. The derivative of $\text{Sig}^{\tilde{\zeta}}(E)$ is given by $\frac{d}{dt} \text{Sig}^{\tilde{\zeta}}(E) = \tilde{\zeta} |E|^{\tilde{\zeta}-1} \dot{E}$. In addition, $\text{diag}(\cdot)$ represents the diagonal matrix, and $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalues, respectively. Additionally, $\|\cdot\|$ denotes the Euclidean norm.

2.1. Dynamic Model of Manipulator

The dynamic robotic model is represented as [1]:

$$M(\alpha)\ddot{\alpha} + C(\alpha, \dot{\alpha})\dot{\alpha} + G(\alpha) = \tau + d. \quad (1)$$

Here, α , $\dot{\alpha}$, and $\ddot{\alpha} \in \mathbb{R}^{n \times 1}$ denote the joint angle position, angular velocity, and angular acceleration, respectively. The inertia matrix is given by $M(\alpha) = \bar{M}(\alpha) + M_{\Delta}(\alpha) \in \mathbb{R}^{n \times n}$, where $\bar{M}(\alpha)$ and $M_{\Delta}(\alpha)$ represent the nominal and uncertain terms of the inertia matrix. Similarly, the centrifugal and Coriolis force matrix is denoted as $C(\alpha, \dot{\alpha}) = \bar{C}(\alpha, \dot{\alpha}) + C_{\Delta}(\alpha, \dot{\alpha}) \in \mathbb{R}^{n \times n}$, with $\bar{C}(\alpha, \dot{\alpha})$ and $C_{\Delta}(\alpha, \dot{\alpha})$ representing the nominal and uncertain terms, respectively. The gravity vector is denoted as $G(\alpha) = \bar{G}(\alpha) + G_{\Delta}(\alpha) \in \mathbb{R}^{n \times 1}$, where $\bar{G}(\alpha)$

and $G_{\Delta}(\alpha)$ represent the nominal and uncertain terms of the gravity vector. The external disturbance is represented by the vector $d \in \mathbb{R}^{n \times 1}$, while the control input torque is denoted as $\tau \in \mathbb{R}^{n \times 1}$. It is important to note that $M_{\Delta}(\alpha)$, $C_{\Delta}(\alpha, \dot{\alpha})$, and $G_{\Delta}(\alpha)$ correspond to the uncertain terms in the dynamic model, with their exact values not known in advance.

Property 1. In Equation (1), M is bounded, symmetric, positive definite, and possesses the following property:

$$\lambda_{\min}(M)\|s\|^2 \leq s^T M s \leq \lambda_{\max}(M)\|s\|^2. \quad (2)$$

Here, $s \in \mathbb{R}^{n \times 1}$ represents any nonzero vector.

Property 2. The following passivity property can be obtained:

$$s^T (\dot{M} - 2C)s = 0, \quad (3)$$

where s represents a vector in $\mathbb{R}^{n \times 1}$.

Let $\alpha_1 = \alpha$, $\alpha_2 = \dot{\alpha}$, and $\dot{\alpha}_2 = \ddot{\alpha}$. Equation (1) can be rewritten in a form suitable for the control design, as follows:

$$\dot{\alpha}_2 = \bar{M}^{-1}(\alpha_1)(-\bar{C}(\alpha_1, \alpha_2)\alpha_2 - \bar{G}(\alpha_1) + \mathcal{F} + \tau). \quad (4)$$

where $\mathcal{F} = -M_{\Delta}(\alpha_1)\dot{\alpha}_2 - C_{\Delta}(\alpha_1, \alpha_2)\alpha_2 - G_{\Delta}(\alpha_1) + d \in \mathbb{R}^{n \times 1}$ represents the uncertain term vector.

This paper addresses challenges in controlling robot manipulators by improving response time, minimizing tracking errors and chattering, and increasing controller robustness. It achieves these objectives by designing a fixed-time sliding mode function based on PPCA, utilizing an RBFNN to approximate the robot's dynamics and handle uncertainty, and developing a novel fixed-time TSMC method. The approach eliminates the need for the robot's dynamical model while ensuring global fixed-time convergence and prescribed performance.

2.2. Preliminaries

A typical representation of a nonlinear system is stated according to Ref. [44]:

$$\dot{s}(t) = f(s(t)), \quad s(t_0) = s_0, \quad s \in \mathbb{R}^n. \quad (5)$$

Here, $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes a possibly discontinuous vector field. System (5) exhibits fixed-time convergence, which is characterized by its global finite-time stability. This property ensures that the system's convergence time remains bounded regardless of the initial states. In other words, for any $s_0 \in \mathbb{R}^n$, the condition $\mathfrak{T}(s_0) \leq \mathfrak{T}_{\max}$ is satisfied, where $\mathfrak{T}_{\max}$ is a positive constant.

Lemma 1 ([44]). Assuming the existence of a positive definite continuous function $V(s) : \mathbb{R}^n \rightarrow \mathbb{R}$ for the system (5), if condition $\dot{V}(s) \leq -\vartheta_1 V^{\chi}(s) - \vartheta_2 V^v(s)$ holds, where $\vartheta_1 > 0$, $\vartheta_2 > 0$, $\chi > 1$, and $0 < v < 1$, then system (5) is globally recognized as a fixed-time stable system. The convergence time of system (5) can be determined independently of the initial conditions using the inequality, presented as follows:

$$\mathfrak{T}(s_0) \leq \frac{1}{\vartheta_1(\chi - 1)} + \frac{1}{\vartheta_2(1 - v)}. \quad (6)$$

Lemma 2 ([44]). If we have a positive definite continuous function $V(s) : \mathbb{R}^n \rightarrow \mathbb{R}$ for system (5), which satisfies $\dot{V}(s) \leq -\vartheta_1 V^{\varkappa}(s) - \vartheta_2 V^{\delta}(s) + \hbar$, where $\vartheta_1 > 0$, $\vartheta_2 > 0$, $\varkappa > 1$, $0 < \delta < 1$, and

$0 < \hbar < \infty$, then system (5) is referred to as a practically fixed-time stable system. Moreover, the solution of system (5) has a residual set:

$$\lim_{t \rightarrow \mathfrak{S}} s \mid \|s\| \leq \min \left\{ \vartheta_1^{-\frac{1}{\varkappa}} \left(\frac{\hbar}{1-\varrho} \right)^{\frac{1}{\varkappa}}, \vartheta_2^{-\frac{1}{\delta}} \left(\frac{\hbar}{1-\varrho} \right)^{\frac{1}{\delta}} \right\}, \quad (7)$$

where ϱ satisfies $0 < \varrho < 1$. The settling time can be calculated independently of the initial states of the system using the following inequality:

$$\mathfrak{S}(s_0) \leq \frac{1}{\vartheta_1 \varrho (1 - \varkappa)} + \frac{1}{\vartheta_2 \varrho (\delta - 1)}. \quad (8)$$

Lemma 3 ([45]). For $s_i \geq 0$, $\varkappa > 1$, and $0 < \delta < 1$, the following inequalities hold:

$$\begin{cases} n^{1-\varkappa} \left(\sum_{i=1}^n s_i \right)^{\varkappa} \leq \sum_{i=1}^n s_i^{\varkappa} \\ \left(\sum_{i=1}^n s_i \right)^{\delta} \leq \sum_{i=1}^n s_i^{\delta} \end{cases}. \quad (9)$$

3. A Model-Based TSMC Approach Using PPCA

3.1. PPCA

The error variable for the position and velocity are defined as follows: $e_{1i} = \alpha_{1di} - \alpha_{1i}$ and $e_{2i} = \alpha_{2di} - \alpha_{2i}$, where $i = 1, \dots, n$. Additionally, α_{1di} is the i^{th} desired position trajectory for the i^{th} joint of the robot. It is desired that the position error e_{1i} at robot joint i remains within a predetermined range:

$$-\ell_b(t) < e_{1i} \text{sign}(e_{1i_0}) < \ell_t(t). \quad (10)$$

Here, $\mathbf{e}$ represents Euler's number, e_{1i_0} is the initial value of e_{1i} at $t = 0$, $\ell_t(t) = (\ell_0 - \ell_\infty)\mathbf{e}^{-\phi t} + \ell_\infty$, and $\ell_b(t) = (\ell_1 - \ell_\infty)\mathbf{e}^{-\phi t} + \ell_\infty$. These functions are smooth, decreasing, positive functions that map $\mathbb{R}^+$ to $\mathbb{R}^+$ and satisfy the conditions $\lim_{t \rightarrow \infty} \ell_t(t) = \ell_\infty > 0$ and $\lim_{t \rightarrow \infty} \ell_b(t) = \ell_\infty > 0$. The constants ℓ_0 , ℓ_1 , and ℓ_∞ are chosen, such that $\ell_0 > |e_{1i_0}| > 0$ and $\ell_0 \geq \ell_1 \geq \ell_\infty$. The positive constant ϕ is utilized to adjust the performance bounds.

The position error can be transformed using the proposed error transformation function (ETF) as follows:

$$\begin{cases} e_{1i} = \mathcal{P}_i(t) \mathcal{T}_i(E_{1i}) \\ \mathcal{T}_i(E_{1i}) = \frac{2}{\pi} \arctan(E_{1i}) \end{cases}, \quad i = 1, \dots, n. \quad (11)$$

Here, E_{1i} represents the i th transformed error.

$$\mathcal{P}_i(t) = \begin{cases} \ell_t(t) & \text{if } \text{sign}(e_{1i} \cdot e_{1i_0}) > 0 \\ \ell_b(t) & \text{if } \text{sign}(e_{1i} \cdot e_{1i_0}) < 0 \end{cases}.$$

The function $\mathcal{T}_i(E_{1i})$ exhibits the following characteristics: it is a smooth and strictly increasing function, satisfying $-1 < \mathcal{T}_i(E_{1i}) < 1$. Additionally, $\mathcal{T}_i(E_{1i}) = 0$ when $E_{1i} = 0$, and as E_{1i} approaches negative infinity, $\lim_{E_{1i} \rightarrow -\infty} \mathcal{T}_i(E_{1i}) = -1$, while as E_{1i} approaches positive infinity, $\lim_{E_{1i} \rightarrow +\infty} \mathcal{T}_i(E_{1i}) = 1$.

If $e_{1i_0} > 0$ and $e_{1i} > 0$, then we have $0 \leq \mathcal{T}_i(E_{1i}) < 1$ and $\ell_t(t) > 0$. This implies that $0 \leq \ell_t(t) \mathcal{T}_i(E_{1i}) < \ell_t(t)$.

On the other hand, if $e_{1i_0} > 0$ and $e_{1i} < 0$, then $\mathcal{T}_i(E_{1i})$ satisfies $-1 < \mathcal{T}_i(E_{1i}) \leq 0$, and $\ell_b(t) > 0$. Consequently, we have $-\ell_b(t) < \ell_b(t)\mathcal{T}_i(E_{1i}) \leq 0$. Thus, when $e_{1i_0} > 0$, the position error e_{1i} falls within the range $-\ell_b(t) < e_{1i} < \ell_t(t)$.

Similarly, if $e_{1i_0} < 0$ and $e_{1i} < 0$, then $\mathcal{T}_i(E_{1i})$ satisfies $-\ell_t(t) < \mathcal{T}_i(E_{1i}) < 0$. Furthermore, if $e_{1i_0} < 0$ and $e_{1i} > 0$, then $\mathcal{T}_i(E_{1i})$ falls within $0 < \mathcal{T}_i(E_{1i}) < \ell_b(t)$. In both cases, we can conclude that when $e_{1i_0} < 0$, the position error e_{1i} lies between $-\ell_t(t)$ and $\ell_b(t)$. Consequently, by utilizing Equation (10), we can define the transformation of the position error in advance, encompassing both the transient and steady-state stages.

From Equation (11), we can obtain E_{1i} and its derivative as follows:

$$E_{1i} = \tan\left(\frac{\pi e_{1i}}{2\mathcal{P}_i(t)}\right), \quad (12)$$

$$\begin{aligned} \dot{E}_{1i} &= \frac{\pi(1+E_{1i}^2)}{2\mathcal{P}_i(t)} \left(e_{2i} - \frac{2\dot{\mathcal{P}}_i(t)}{\pi} \arctan(E_{1i}) \right) \\ &= A_i(e_{2i} - \wp_i) \end{aligned} \quad (13)$$

Here, we define $A = \text{diag}(A_i)$ with $A_i = \frac{\pi(1+E_{1i}^2)}{2\mathcal{P}_i(t)} > 0$ and $\wp = [\wp_1 \ \dots \ \wp_n]^T$ with $\wp_i = \frac{2\dot{\mathcal{P}}_i(t)}{\pi} \arctan(E_{1i})$, for $i = 1, \dots, n$.

Let $E_{2i} = e_{2i} - \wp_i$. Thus, Equation (13) can be rewritten as follows:

$$\begin{cases} \dot{E}_{1i} = A_i E_{2i} \\ \dot{E}_{2i} = \dot{e}_{2i} - \mathcal{H}_i \end{cases}, \quad i = 1, \dots, n. \quad (14)$$

Here, we define $\mathcal{H} = [\mathcal{H}_1 \ \dots \ \mathcal{H}_n]^T$ with $\mathcal{H}_i = \dot{\wp}_i = \frac{2}{\pi} \left(\dot{\mathcal{P}}_i(t) \arctan(E_{1i}) + \mathcal{P}_i(t) \frac{\dot{E}_{1i}}{1+E_{1i}^2} \right)$, for $i = 1, \dots, n$.

3.2. Design of a Prescribed Performance Sliding Surface

We introduce a prescribed performance sliding surface as follows:

$$S_i = E_{2i} + \kappa_1 \text{Sig}^{\xi_1}(E_{1i}) + \kappa_2 \text{Sig}^{\xi_2}(E_{1i}), \quad i = 1, \dots, n, \quad (15)$$

where $\kappa_1 > 0, \kappa_2 > 0, \xi_1 > 1$, and $\frac{1}{2} < \xi_2 < 1$.

Once $S_i = 0$ for $i = 1, \dots, n$, we can derive the following expression:

$$\dot{E}_{1i} \leq -\mathcal{K}_1 \text{Sig}^{\xi_1}(E_{1i}) - \mathcal{K}_2 \text{Sig}^{\xi_2}(E_{1i}). \quad (16)$$

where $\mathcal{K}_1 = A_{i\min}\kappa_1, \mathcal{K}_2 = A_{i\min}\kappa_2$, and $A_{i\min} = \frac{\pi}{2\ell_{0i}}$.

Similar to Lemma 1, the introduced sliding surface in Equation (15) exhibits the same characteristics, resulting in fixed-time convergence with a specific settling time denoted as $\mathfrak{S}(E_{i_0}) < \mathfrak{S}_{\max}$. The maximum settling time is given by the expression $\frac{1}{\mathcal{K}_1(\xi_1-1)} + \frac{1}{\mathcal{K}_2(1-\xi_2)}$.

3.3. Design of Model-Based TSMC Method Using PPCA

Substituting Equation (13) into Equation (15) yields:

$$\begin{aligned} -(e_{2i} - \wp_i) &= \mathcal{K}_1 \text{Sig}^{\xi_1}(E_{1i}) + \mathcal{K}_2 \text{Sig}^{\xi_2}(E_{1i}) - S_i \\ -e_{2i} &= \mathcal{K}_1 \text{Sig}^{\xi_1}(E_{1i}) + \mathcal{K}_2 \text{Sig}^{\xi_2}(E_{1i}) - S_i - \wp_i \\ (\alpha_{2i} - \alpha_{2di}) &= \mathcal{K}_1 \text{Sig}^{\xi_1}(E_{1i}) + \mathcal{K}_2 \text{Sig}^{\xi_2}(E_{1i}) - S_i - \wp_i \\ \alpha_{2i} &= \mathcal{K}_1 \text{Sig}^{\xi_1}(E_{1i}) + \mathcal{K}_2 \text{Sig}^{\xi_2}(E_{1i}) - S_i - \wp_i + \alpha_{2di} \end{aligned} \quad (17)$$

We define $\mathcal{V} = [\mathcal{V}_1 \dots \mathcal{V}_n]^T$ with $\mathcal{V}_i = \mathcal{K}_1 \text{Sig}^{\xi_1}(E_{1i}) + \mathcal{K}_2 \text{Sig}^{\xi_2}(E_{1i}) - \wp_i + \alpha_{2di}$, for $i = 1, \dots, n$. Therefore, Equation (17) can be written as $\alpha_{2i} = \mathcal{V}_i - S_i$, and its vector form is as follows:

$$\alpha_2 = \mathcal{V} - S. \quad (18)$$

With respect to time, taking the derivative of Equation (15) yields the following results:

$$\dot{S}_i = \dot{E}_{2i} + \mathcal{K}_1 \xi_1 |E_{1i}|^{\xi_1-1} \dot{E}_{1i} + \mathcal{K}_2 \xi_2 |E_{1i}|^{\xi_2-1} \dot{E}_{1i}. \quad (19)$$

Substituting Equation (14) into Equation (19) yields:

$$\begin{aligned} \dot{S}_i &= (\dot{e}_{2i} - \mathcal{H}_i) + \mathcal{K}_1 \xi_1 |E_{1i}|^{\xi_1-1} \dot{E}_{1i} + \mathcal{K}_2 \xi_2 |E_{1i}|^{\xi_2-1} \dot{E}_{1i} \\ \dot{S}_i &= \dot{e}_{2i} + \mathcal{K}_1 \xi_1 |E_{1i}|^{\xi_1-1} \dot{E}_{1i} + \mathcal{K}_2 \xi_2 |E_{1i}|^{\xi_2-1} \dot{E}_{1i} - \mathcal{H}_i \\ \dot{S}_i &= (\dot{\alpha}_{2di} - \dot{\alpha}_{2i}) + \mathcal{K}_1 \xi_1 |E_{1i}|^{\xi_1-1} \dot{E}_{1i} + \mathcal{K}_2 \xi_2 |E_{1i}|^{\xi_2-1} \dot{E}_{1i} - \mathcal{H}_i \\ \dot{S}_i &= \dot{\alpha}_{2di} + \mathcal{K}_1 \xi_1 |E_{1i}|^{\xi_1-1} \dot{E}_{1i} + \mathcal{K}_2 \xi_2 |E_{1i}|^{\xi_2-1} \dot{E}_{1i} - \mathcal{H}_i - \dot{\alpha}_{2i} \end{aligned} \quad (20)$$

By defining $\mathcal{Y} = [\mathcal{Y}_1 \dots \mathcal{Y}_n]^T$ with $\mathcal{Y}_i = \dot{\alpha}_{2di} + \mathcal{K}_1 \xi_1 |E_{1i}|^{\xi_1-1} \dot{E}_{1i} + \mathcal{K}_2 \xi_2 |E_{1i}|^{\xi_2-1} \dot{E}_{1i} - \mathcal{H}_i$, for $i = 1, \dots, n$, Equation (20) can be written as $\dot{S}_i = \mathcal{Y}_i - \dot{\alpha}_{2i}$, and its vector form is as follows:

$$\dot{S} = \mathcal{Y} - \dot{\alpha}_2. \quad (21)$$

By multiplying both sides of Equation (21) with $\bar{M}(\alpha_1)$ and applying Equations (4) and (18), the following expression is derived:

$$\begin{aligned} \bar{M}(\alpha_1) \dot{S} &= \bar{M}(\alpha_1) (\mathcal{Y} - \dot{\alpha}_2) \\ &= \bar{M}(\alpha_1) \mathcal{Y} - \bar{M}(\alpha_1) \dot{\alpha}_2 \\ &= \bar{M}(\alpha_1) \mathcal{Y} - \bar{M}(\alpha_1) (\bar{M}^{-1}(\alpha_1) (-\bar{C}(\alpha_1, \alpha_2) \alpha_2 - \bar{G}(\alpha_1) + \mathcal{F} + \tau)) \\ &= \bar{M}(\alpha_1) \mathcal{Y} + \bar{C}(\alpha_1, \alpha_2) \alpha_2 + \bar{G}(\alpha_1) - \mathcal{F} - \tau \\ &= \bar{M}(\alpha_1) \mathcal{Y} + \bar{C}(\alpha_1, \alpha_2) (\mathcal{V} - S) + \bar{G}(\alpha_1) - \mathcal{F} - \tau \\ &= \bar{M}(\alpha_1) \mathcal{Y} + \bar{C}(\alpha_1, \alpha_2) \mathcal{V} - \bar{C}(\alpha_1, \alpha_2) S + \bar{G}(\alpha_1) - \mathcal{F} - \tau \end{aligned} \quad (22)$$

A Lyapunov function is considered as follows:

$$V_0 = \frac{1}{2} S^T \bar{M}(\alpha_1) S. \quad (23)$$

By differentiating V_0 in Equation (23) with respect to time, we obtain the following expression:

$$\begin{aligned} \dot{V}_0 &= S^T \bar{M}(\alpha_1) \dot{S} + \frac{1}{2} S^T \dot{\bar{M}}(\alpha_1) S \\ &= S^T (\bar{M}(\alpha_1) \mathcal{Y} + \bar{C}(\alpha_1, \alpha_2) \mathcal{V} - \bar{C}(\alpha_1, \alpha_2) S + \bar{G}(\alpha_1) - \mathcal{F} - \tau) + \frac{1}{2} S^T \dot{\bar{M}}(\alpha_1) S \\ &= S^T (\bar{M}(\alpha_1) \mathcal{Y} + \bar{C}(\alpha_1, \alpha_2) \mathcal{V} + \bar{G}(\alpha_1) - \mathcal{F} - \tau) + \frac{1}{2} S^T (\dot{\bar{M}}(\alpha_1) - 2\bar{C}(\alpha_1, \alpha_2)) S \end{aligned} \quad (24)$$

Referring to Property 2, we have $S^T (\dot{\bar{M}}(\alpha_1) - 2\bar{C}(\alpha_1, \alpha_2)) S = 0$. Consequently, we obtain:

$$\dot{V}_0 = S^T (\bar{M}(\alpha_1) \mathcal{Y} + \bar{C}(\alpha_1, \alpha_2) \mathcal{V} + \bar{G}(\alpha_1) - \mathcal{F} - \tau). \quad (25)$$

Based on Equation (25), the control law is formulated as follows:

$$\begin{cases} \tau = \tau_{eq} + \tau_r \\ \tau_{eq} = \bar{M}(\alpha_1) \mathcal{Y} + \bar{C}(\alpha_1, \alpha_2) \mathcal{V} + \bar{G}(\alpha_1) - \mathcal{F} \\ \tau_r = \gamma \text{sign}(S) + \psi_1 \text{Sig}^{2\omega_1-1}(S) + \psi_2 \text{Sig}^{2\omega_2-1}(S) \end{cases} \quad (26)$$

In the control law, two terms are considered: τ_{eq} and τ_r . The term τ_{eq} is designed based on the mathematical model of the robotic manipulator, while the term τ_r is designed as a

fixed-time reaching control law. The constants ψ_1 , ψ_2 , and γ are positive, with $\omega_1 > 1$ and $\frac{1}{2} < \omega_2 < 1$.

A summary of the control design is stated in the following theorem.

Theorem 1. *Assuming that we have knowledge of the dynamic model of the robotic manipulator, including $\bar{M}(\alpha_1)$, $\bar{C}(\alpha_1, \alpha_2)$, $\bar{G}(\alpha_1)$, and $\mathcal{F}$, we can design the control law as in Equation (26) to achieve the prescribed performance for the robot in a fixed time.*

Proof of Theorem 1. By applying the control law (26) to Equation (25), we obtain:

$$\begin{aligned}\dot{V}_0 &= S^T \left(-\gamma \text{sign}(S) - \psi_1 \text{Sig}^{2\omega_1-1}(S) - \psi_2 \text{Sig}^{2\omega_2-1}(S) \right) \\ &= -S^T \gamma \text{sign}(S) - S^T \psi_1 \text{Sig}^{2\omega_1-1}(S) - S^T \psi_2 \text{Sig}^{2\omega_2-1}(S) \\ &\leq -\lambda_{\min}(\gamma) \|S\| - S^T \psi_1 \text{Sig}^{2\omega_1-1}(S) - S^T \psi_2 \text{Sig}^{2\omega_2-1}(S) \\ &\leq -S^T \psi_1 \text{Sig}^{2\omega_1-1}(S) - S^T \psi_2 \text{Sig}^{2\omega_2-1}(S)\end{aligned}\quad (27)$$

Rewriting Equation (27) and using Property 1 and Lemma 3, we can obtain the following expression:

$$\begin{aligned}\dot{V}_0 &\leq -\sum_{i=1}^n \psi_{1i} |S_i|^{2\omega_1} - \sum_{i=1}^n \psi_{2i} |S_i|^{2\omega_2} \\ &\leq -\lambda_{\min}(\psi_1) 2^{\omega_1} n^{1-\omega_1} \left(\frac{1}{2} \sum_{i=1}^n S_i^2 \right)^{\omega_1} - \lambda_{\min}(\psi_2) 2^{\omega_2} \left(\frac{1}{2} \sum_{i=1}^n S_i^2 \right)^{\omega_2} \\ &\leq -\lambda_{\min}(\psi_1) 2^{\omega_1} n^{1-\omega_1} \left(\frac{1}{2} \|S\|^2 \right)^{\omega_1} - \lambda_{\min}(\psi_2) 2^{\omega_2} \left(\frac{1}{2} \|S\|^2 \right)^{\omega_2} \\ &\leq -\frac{\lambda_{\min}(\psi_1)}{\lambda_{\max}(M)^{\omega_1}} 2^{\omega_1} n^{1-\omega_1} \left(\frac{1}{2} S^T M S \right)^{\omega_1} - \frac{\lambda_{\min}(\psi_2)}{\lambda_{\max}(M)^{\omega_2}} 2^{\omega_2} \left(\frac{1}{2} S^T M S \right)^{\omega_2} \\ &\leq -\eta_1 (V_0)^{\omega_1} - \eta_2 (V_0)^{\omega_2}\end{aligned}\quad (28)$$

where $\eta_1 = \frac{\lambda_{\min}(\psi_1)}{\lambda_{\max}(M)^{\omega_1}} 2^{\omega_1} n^{1-\omega_1}$ and $\eta_2 = \frac{\lambda_{\min}(\psi_2)}{\lambda_{\max}(M)^{\omega_2}} 2^{\omega_2}$. $\square$

In practical scenarios, computing the dynamic model of a robot manipulator can be a challenging task due to several factors. These factors include the intricate mechanical structure of the robot, the existence of multiple degrees of freedom, and the variability in the payload. In situations where the mathematical model of the robot is unknown or challenging to determine, an RBFNN can be employed for estimation purposes. This allows the function $f(s) = \bar{M}(\alpha_1)\mathcal{Y} + \bar{C}(\alpha_1, \alpha_2)\mathcal{Y} + \bar{G}(\alpha_1) - \mathcal{F}$ to be approximated by the RBFNN, enabling the prediction of the robot's behavior. As a result, model-free control schemes can be formulated based on the estimated model.

4. A Model-Free TSMC Approach Using RBFNN and PPCA

4.1. RBFNN

An RBFNN can always accurately approximate the function $f(s)$, as demonstrated below:

$$\mathcal{N}(\mathcal{X}) = f(s) \text{ with } \mathcal{N}(\mathcal{X}) = \mathcal{W}^T \mathcal{G}(\mathcal{X}) + \iota. \quad (29)$$

The vector $\mathcal{X} = [E_1^T, E_2^T, \alpha_{1d}^T, \alpha_{2d}^T, \dot{\alpha}_{2d}^T]^T \in \mathbb{R}^{5n \times 1}$ represents the input of the RBFNN and $\mathcal{W} \in \mathbb{R}^{l \times n}$ represents the optimal weight. The weight of the NN is achieved by optimizing the approximation error over the training set:

$$\mathcal{W} = \arg \min_{\mathcal{W} \in \mathbb{R}^l} \left\{ \sup_{\mathcal{X} \in \Omega_{\mathcal{X}}} |\mathcal{N}(\mathcal{X}) - \mathcal{W}^T \mathcal{G}(\mathcal{X})| \right\}. \quad (30)$$

Here, $\iota \in \mathbb{R}^{n \times 1}$ represents the approximation error, and $|\iota| \leq \gamma^*$, where $\gamma^* \geq 0$ [34]. The function of the hidden nodes is $\mathcal{G}(\mathcal{X})$, which is chosen as the Gaussian function described below:

$$\mathcal{G}(\mathcal{X}) = \exp\left(\frac{-(\mathcal{X} - \sigma_i)^T(\mathcal{X} - \sigma_i)}{\mathcal{D}_i^2}\right), \quad (31)$$

$$i = 1, 2, \dots, l.$$

In Equation (31), the width and center of $\mathcal{G}(\mathcal{X})$ are represented by $\mathcal{D}$ and σ , respectively, where l denotes the number of hidden nodes.

The network produces the following outputs:

$$\hat{\mathcal{N}}(\mathcal{X}) = \hat{\mathcal{W}}^T \mathcal{G}(\mathcal{X}) \in \mathbb{R}^{n \times 1}, \quad (32)$$

where the estimation of the optimal weight is represented by $\hat{\mathcal{W}}$.

Let $\tilde{\mathcal{W}} = \mathcal{W} - \hat{\mathcal{W}}$ be the estimation error of the optimal weights. Therefore,

$$\begin{aligned} \mathcal{N}(\mathcal{X}) - \hat{\mathcal{N}}(\mathcal{X}) &= \mathcal{W}^T \mathcal{G}(\mathcal{X}) + \iota - \hat{\mathcal{W}}^T \mathcal{G}(\mathcal{X}) \\ &= \tilde{\mathcal{W}}^T \mathcal{G}(\mathcal{X}) + \iota. \end{aligned} \quad (33)$$

4.2. Design of Model-Free TSMC Method Using RBFNN and PPCA

The control law is expressed in the following manner:

$$\begin{cases} \tau = \hat{f}(s) + \tau_r \\ \hat{f}(s) = \hat{\mathcal{W}}^T \mathcal{G}(\mathcal{X}) \\ \tau_r = \gamma^* \text{sign}(S) + \psi_1 \text{Sig}^{2\omega_1-1}(S) + \psi_2 \text{Sig}^{2\omega_2-1}(S) \end{cases}. \quad (34)$$

In the control law, $\hat{f}(s)$ and τ_r represent two distinct terms. $\hat{f}(s)$ is obtained from the output of the NN, while the term τ_r is designed as a fixed-time reaching control law. The constants ψ_1 and ψ_2 are positive, with $\omega_1 > 1$, $\frac{1}{2} < \omega_2 < 1$, and $\gamma^* > |\iota|$.

We design update laws for the NN to achieve fixed-time stability for both the tracking error and the neural network itself, as follows:

$$\dot{\hat{\mathcal{W}}}_i = \mathcal{K}_N^{-1} \left(\mathcal{G}(\mathcal{X}) S_i - \psi_3 \hat{\mathcal{W}}_i^{2\omega_1-1} - \psi_4 \hat{\mathcal{W}}_i^{2\omega_2-1} \right). \quad (35)$$

Here, $\hat{\mathcal{W}}_i$ represents the i th element of the matrix $\mathcal{W}$.

A summary of the control design is stated in the following theorem, and the control diagram is shown in Figure 1.

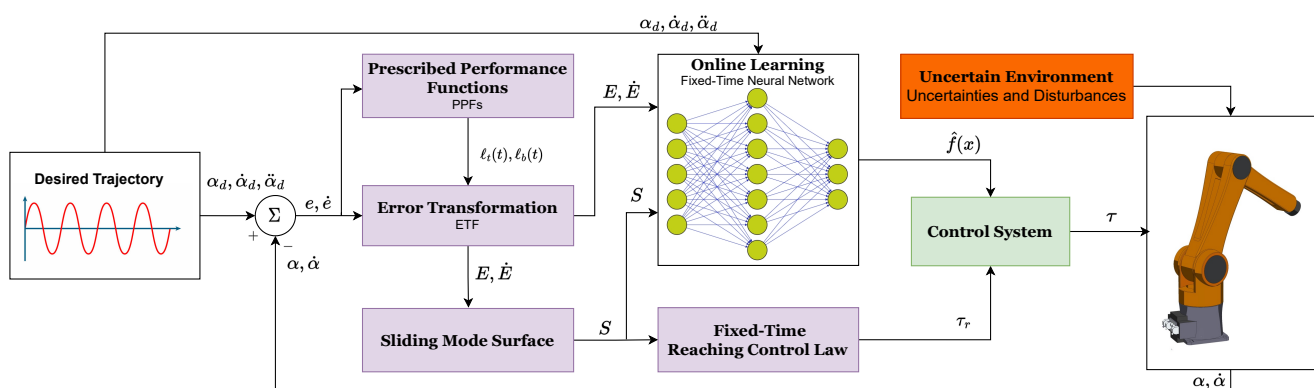


Figure 1. Graph of the control system.

Theorem 2. By employing an RBFNN with update laws (35), we can estimate the unknown function $f(s)$, which is associated with the robot's dynamics and dynamic uncertainties. Utilizing the control law defined in Equation (34), we can achieve both the desired performance for the robot's tracking errors and the convergence of the estimation errors of the NN within a fixed time.

Proof of Theorem 2. Substituting Equation (34) into Equation (22) yields:

$$\begin{aligned}\bar{M}(\alpha_1)\dot{S} &= f(s) - \bar{C}(\alpha_1, \alpha_2)S - \hat{f}(s) - \tau_r \\ &= \tilde{W}^T \mathcal{G}(\mathcal{X}) + \iota - \bar{C}(\alpha_1, \alpha_2)S - \tau_r.\end{aligned}\quad (36)$$

The Lyapunov function is selected as follows:

$$V_1 = \frac{1}{2} S^T \bar{M}(\alpha_1) S. \quad (37)$$

Computing the time derivative of V_1 using the result of Equation (36), we obtain:

$$\begin{aligned}\dot{V}_1 &= S^T \bar{M}(\alpha_1) \dot{S} + \frac{1}{2} S^T \dot{\bar{M}}(\alpha_1) S \\ &= S^T (\tilde{W}^T \mathcal{G}(\mathcal{X}) + \iota - \bar{C}(\alpha_1, \alpha_2)S - \tau_r) + \frac{1}{2} S^T \dot{\bar{M}}(\alpha_1) S \\ &= (S^T \tilde{W}^T \mathcal{G}(\mathcal{X}) + S^T \iota - S^T \bar{C}(\alpha_1, \alpha_2)S - S^T \tau_r) + \frac{1}{2} S^T \dot{\bar{M}}(\alpha_1) S \\ &= S^T \tilde{W}^T \mathcal{G}(\mathcal{X}) + S^T \iota - S^T \tau_r + 0.5 S^T (\dot{\bar{M}}(\alpha_1) - 2\bar{C}(\alpha_1, \alpha_2)) S.\end{aligned}\quad (38)$$

Since $S^T (\dot{\bar{M}}(\alpha_1) - 2\bar{C}(\alpha_1, \alpha_2)) S = 0$, we can subsequently obtain:

$$\begin{aligned}\dot{V}_1 &= S^T \tilde{W}^T \mathcal{G}(\mathcal{X}) + S^T \iota - S^T \tau_r \\ &= S^T \tilde{W}^T \mathcal{G}(\mathcal{X}) - S^T \gamma^* \text{sign}(S) - S^T \psi_1 \text{Sig}^{2\omega_1-1}(S) - S^T \psi_2 \text{Sig}^{2\omega_2-1}(S) + S^T \iota \\ &\leq S^T \tilde{W}^T \mathcal{G}(\mathcal{X}) - S^T \psi_1 \text{Sig}^{2\omega_1-1}(S) - S^T \psi_2 \text{Sig}^{2\omega_2-1}(S)\end{aligned}\quad (39)$$

Like Equations (27) and (28), we can obtain the following equality:

$$\dot{V}_1 \leq S^T \tilde{W}^T \mathcal{G}(\mathcal{X}) - \eta_1 (V_1)^{\omega_1} - \eta_2 (V_1)^{\omega_2}, \quad (40)$$

where $\eta_1 = \frac{\lambda_{\min}(\psi_1)}{\lambda_{\max}(M)^{\omega_1}} 2^{\omega_1} n^{1-\omega_1}$ and $\eta_2 = \frac{\lambda_{\min}(\psi_2)}{\lambda_{\max}(M)^{\omega_2}} 2^{\omega_2}$.

Select a Lyapunov function for the convergence of RBFNN as follows:

$$V_2 = \frac{1}{2} \mathcal{K}_N \sum_{i=1}^n \tilde{W}_i^T \tilde{W}_i. \quad (41)$$

By considering update law (35), we can derive the following expression for the derivative of Equation (41):

$$\begin{aligned}\dot{V}_2 &= -\mathcal{K}_N \sum_{i=1}^n \tilde{W}_i^T \dot{\tilde{W}}_i \\ &= -\sum_{i=1}^n \tilde{W}_i^T (\mathcal{G}(\mathcal{X}) S_i - \psi_3 \hat{W}_i^{2\omega_1-1} - \psi_4 \hat{W}_i^{2\omega_2-1}) \\ &= -\sum_{i=1}^n \tilde{W}_i^T \mathcal{G}(\mathcal{X}) S_i + \psi_3 \sum_{i=1}^n \tilde{W}_i^T \hat{W}_i^{2\omega_1-1} + \psi_4 \sum_{i=1}^n \tilde{W}_i^T \hat{W}_i^{2\omega_2-1}\end{aligned}\quad (42)$$

By utilizing the inequality:

$$\begin{cases} \tilde{w}_{ij} \hat{w}_{ij}^{2\omega_1-1} \leq w_{ij}^{2\omega_1} - \tilde{w}_{ij}^{2\omega_1} \\ \tilde{w}_{ij} \hat{w}_{ij}^{2\omega_2-1} \leq v_1 w_{ij}^{2\omega_2} - v_2 \tilde{w}_{ij}^{2\omega_2} \end{cases}, \quad (43)$$

where w_{ij} is the j th element of the vector $\mathcal{W}_i$, $i = 1, \dots, n$, and $j = 1, \dots, l$.

Consequently, Equation (42) can be rewritten as follows:

$$\begin{aligned}
 \dot{V}_2 &\leq -\sum_{i=1}^n \tilde{\mathcal{W}}_i^T \mathcal{G}(\mathcal{X}) S_i + \psi_3 \sum_{i=1}^n \sum_{j=1}^l \left(w_{ij}^{2\omega_1} - \tilde{w}_{ij}^{2\omega_1} \right) + \psi_4 \sum_{i=1}^n \sum_{j=1}^l \left(v_1 w_{ij}^{2\omega_2} - v_2 \tilde{w}_{ij}^{2\omega_2} \right) \\
 &= -\sum_{i=1}^n \tilde{\mathcal{W}}_i^T \mathcal{G}(\mathcal{X}) S_i - \psi_3 \sum_{i=1}^n \sum_{j=1}^l \left(\tilde{w}_{ij}^{2\omega_1} \right) - \psi_4 v_2 \sum_{i=1}^n \sum_{j=1}^l \left(\tilde{w}_{ij}^{2\omega_2} \right) + \psi_3 \sum_{i=1}^n \sum_{j=1}^l \left(w_{ij}^{2\omega_1} \right) + \psi_4 v_1 \sum_{i=1}^n \sum_{j=1}^l \left(w_{ij}^{2\omega_2} \right) \\
 &= -\sum_{i=1}^n \tilde{\mathcal{W}}_i^T \mathcal{G}(\mathcal{X}) S_i - \psi_3 \sum_{i=1}^n \sum_{j=1}^l \left(\tilde{w}_{ij}^{2\omega_1} \right) - \psi_4 v_2 \sum_{i=1}^n \sum_{j=1}^l \left(\tilde{w}_{ij}^{2\omega_2} \right) + \hbar \\
 &\leq -\sum_{i=1}^n \tilde{\mathcal{W}}_i^T \mathcal{G}(\mathcal{X}) S_i - \psi_3 l^{1-\omega_1} \sum_{i=1}^n \left(\sum_{j=1}^l \tilde{w}_{ij}^2 \right)^{\omega_1} - \psi_4 v_2 \sum_{i=1}^n \left(\sum_{j=1}^l \tilde{w}_{ij}^2 \right)^{\omega_2} + \hbar \\
 &= -\sum_{i=1}^n \tilde{\mathcal{W}}_i^T \mathcal{G}(\mathcal{X}) S_i - \left(\frac{2}{\kappa_N} \right)^{\omega_1} \psi_3 l^{1-\omega_1} \left(\frac{\kappa_N}{2} \sum_{i=1}^n \tilde{\mathcal{W}}_i^T \tilde{\mathcal{W}}_i \right)^{\omega_1} - \left(\frac{2}{\kappa_N} \right)^{\omega_2} \psi_4 v_2 \left(\frac{\kappa_N}{2} \sum_{i=1}^n \tilde{\mathcal{W}}_i^T \tilde{\mathcal{W}}_i \right)^{\omega_2} + \hbar \\
 &\leq -\sum_{i=1}^n \tilde{\mathcal{W}}_i^T \mathcal{G}(\mathcal{X}) S_i - \eta_3 (V_2)^{\omega_1} - \eta_4 (V_2)^{\omega_2} + \hbar
 \end{aligned} \tag{44}$$

where $\eta_3 = \left(\frac{2}{\kappa_N} \right)^{\omega_1} \psi_3 l^{1-\omega_1}$, $\eta_4 = \left(\frac{2}{\kappa_N} \right)^{\omega_2} \psi_4 v_2$, and $\hbar = \psi_3 \sum_{i=1}^n \sum_{j=1}^l \left(w_{ij}^{2\omega_1} \right) + \psi_4 v_1 \sum_{i=1}^n \sum_{j=1}^l \left(w_{ij}^{2\omega_2} \right)$.

Choose V as the Lyapunov function for the convergence and stability of the entire control system that includes the NN. Hence, we have $V = V_1 + V_2$. From the results of Equations (40) and (44), the derivative of V is given by:

$$\begin{aligned}
 \dot{V} &\leq -\eta_1 (V_1)^{\omega_1} - \eta_2 (V_1)^{\omega_2} - \eta_3 (V_2)^{\omega_1} - \eta_4 (V_2)^{\omega_2} + \hbar \\
 &\leq -\vartheta_1 (V)^{\omega_1} - \vartheta_2 (V)^{\omega_2} + \hbar
 \end{aligned} \tag{45}$$

where $\vartheta_1 = 2^{1-\omega_1} \min(\eta_1, \eta_3)$ and $\vartheta_2 = \min(\eta_2, \eta_4)$.

Therefore, by applying Lemma 2, we can guarantee the global fixed-time convergence of the control system with a settling time bounded by $\mathfrak{S} < \mathfrak{S}_{\max} \triangleq \frac{1}{\vartheta_1 \varrho (\omega_1 - 1)} + \frac{1}{\vartheta_2 \varrho (1 - \omega_2)}$. $\square$

Remark 1. Unlike conventional approaches that use tracking errors as inputs for the network, the proposed network utilizes inputs chosen as $\mathcal{X} = \left[E_1^T, E_2^T, \alpha_{1d}^T, \alpha_{2d}^T, \dot{\alpha}_{2d}^T \right]^T$, minimizing the transformed errors while also constraining the tracking errors. This novel approach leads to a more accurate estimation compared to existing studies.

Remark 2. Unlike references [39,40,42,43], this approach uses two separate PPFs to set bounds for the tracking error. One PPF limits the convergence rate and steady-state error, while the other addresses overshoot and steady-state error. This enlarges the performance space at a steady-state compared to traditional methods. Additionally, the symmetric steady-state error boundaries ensure the zero-tracking error when the transformed error is zero, simplifying the ETF design. Our ETF is also free from singularity problems.

Remark 3. It can be seen that the proposed method has not been applied to real systems yet and requires powerful enough computing hardware to increase the computational speed of the NN. Therefore, our future research direction is to extend the application of the proposed controller to a practical system known as 6-DOF FARA SAMSUNG Robot Manipulator.

5. Simulations

5.1. Testing System Configuration: A Detailed Examination of the Setup

The simulations were performed on a 3-DOF robot using MATLAB/SIMULINK R2021b. The mechanical components of the robot were designed using SOLIDWORKS

2018 and subsequently integrated into the MATLAB/SIMULINK environment using the SIMSCAPE MULTIBODY LINK tool. This ensured that the simulation model of the robot was an accurate representation of the actual mechanical model. Figure 2 displays the robot model, and Table 1 lists the essential design specifications for the system, such as link dimensions, center of mass position, and inertia [13]. These specifications are incorporated into the SOLIDWORKS robot model. The proposed method does not rely on the mathematical model of the robot for control design.

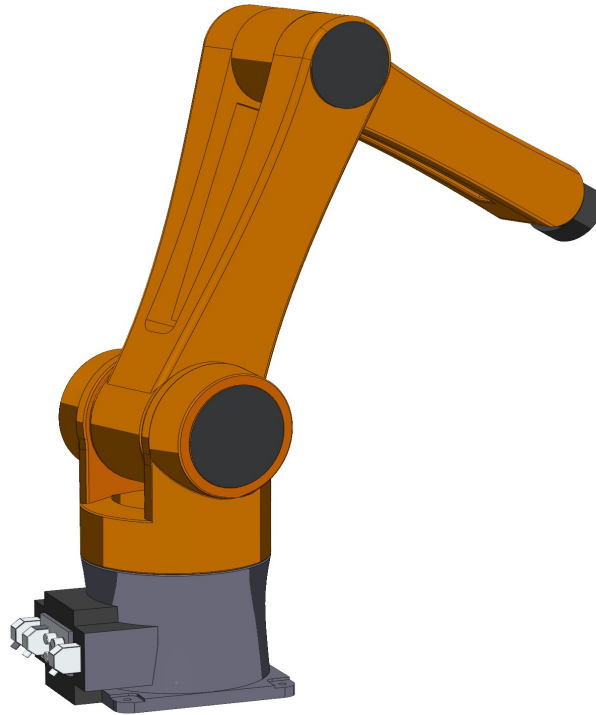


Figure 2. Designed model of a 3-DOF robot used in simulations [13].

Table 1. Essential design specifications for the manipulator.

Description	Link 1	Link 2	Link 3
Length (m)	$l_1 = 0.25$	$l_2 = 0.7$	$l_3 = 0.6$
Weight (kg)	$m_1 = 33.429$	$m_2 = 34.129$	$m_3 = 15.612$
Center of Mass (mm)	$l_{c1x} = 0$ $l_{c1y} = 0$ $l_{c1z} = -0.7461$	$l_{c2x} = 0.3477$ $l_{c2y} = 0$ $l_{c2z} = 0$	$l_{c3x} = 0.3142$ $l_{c3y} = 0$ $l_{c3z} = 0$
Inertia ($\text{kg}\cdot\text{m}^2$)	$I_{1xx} = 0.7486$ $I_{1yy} = 0.5518$ $I_{1zz} = 0.5570$	$I_{2xx} = 0.3080$ $I_{2yy} = 2.4655$ $I_{2zz} = 2.3938$	$I_{3xx} = 0.0446$ $I_{3yy} = 0.7092$ $I_{3zz} = 0.7207$

The robot's main objective is to follow a predefined trajectory accurately using its end-effector, which may involve complex motions and varying speeds. An example of such a desired trajectory is shown in the equation below, which describes the position of the end-effector as a function of time in a three-dimensional space, as shown in Table 2.

Simulations were conducted to validate the proposed system under normal operating conditions of the robot, taking into account uncertainty components, such as friction forces and disturbances, as shown in Table 2. The performance of the developed method was compared with other model-based controllers, including SMC, TSMC, and fast TSMC (FTSMC). Additionally, it was compared to a PID controller, a well-known model-free controller, to demonstrate its superior performance. The objective of these comparisons was to highlight the advantages of the proposed method over existing control approaches.

To ensure fairness, control parameters from the original papers were utilized during the simulation of the controllers, considering the differences in controller structures. Parameters for the proposed controller were determined through experimentation. The design parameters of the developed algorithm are reported in Table 3.

Table 2. Investigating the effectiveness of a proposed control system by tracking the desired trajectory for a robot manipulator with assumed uncertain terms in the simulation.

Terms	Described Function
Desired Trajectory (m)	$X_d = 0.85 - 0.01t$ $Y_d = 0.2 + 0.2 \sin(\frac{t}{2})$ $Z_d = 0.7 + 0.2 \cos(\frac{t}{2})$
Friction Forces and Disturbances $\mathcal{F}_r(\alpha_2) + d(t)$ (N·m)	$\mathcal{F}_{r1}(\alpha_2) + d_1(t) = 0.1\text{sign}(\alpha_{21}) + 2\alpha_{21} + 4 \sin(t)$ $\mathcal{F}_{r2}(\alpha_2) + d_2(t) = 0.1\text{sign}(\alpha_{22}) + 2\alpha_{22} + 5 \sin(t)$ $\mathcal{F}_{r3}(\alpha_2) + d_3(t) = 0.1\text{sign}(\alpha_{23}) + 2\alpha_{23} + 6 \sin(t)$

Table 3. Design parameters of the developed scheme.

Technique	Notation	Value
PPC	$\ell_0, \ell_1, \ell_\infty, \phi$	$[0.34 \quad 0.15 \quad 0.3], 0.06, 0.001, 6$
RBFNN	$\mathcal{K}_N, \psi_3, \psi_4, \mathcal{W}_0$ $\sigma, l, \mathcal{D}$	$\frac{1}{1000}, 0.001, 0.001, \text{zeros}(9,3)$ $0.01 \times \begin{bmatrix} -2 & -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -2 & -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 & 2 \end{bmatrix} \in \mathbb{R}^{9 \times 15}, 9, 10$
Developed Algorithm	$\kappa_1, \kappa_2, \xi_1, \xi_2$ $\psi_1, \psi_2, \omega_1, \omega_2, \gamma^*$	$0.1, 0.1, 1.4, 0.8$ $10, 500, 1.2, 0.8, 0.00001$

Assuming we possess knowledge of all the robot dynamics, including $\bar{M}(\alpha_1)$, $\bar{C}(\alpha_1, \alpha_2)$, and $\bar{G}(\alpha_1)$, with the exception of the uncertain term $\mathcal{F}$, we can proceed to design various model-based controllers for the robot as outlined below.

According to [15], the sliding surface of the SMC is constructed as follows:

$$S = e_2 + \mathcal{C}e_1, \quad (46)$$

where $\mathcal{C} = \text{diag}(c_i)$ with $c_i > 0$ for $i = 1, \dots, n$.

The control torque obtained from the output of the SMC is expressed as follows:

$$\tau = \bar{M}(\alpha_1)(\dot{\alpha}_{2d} + \mathcal{C}e_2 + \Lambda \text{sign}(S) + \Delta S) + \bar{C}(\alpha_1, \alpha_2)\alpha_2 + \bar{G}(\alpha_1), \quad (47)$$

where $\Delta = \text{diag}(\delta_i)$ with $\delta_i > 0$ for $i = 1, \dots, n$, and $|\mathcal{F}| \leq \Lambda$.

In the TSMC approach [46], a sliding function is constructed as follows:

$$S = e_1 + \mathcal{C}^{-1} \text{Sig}^{\frac{p}{q}}(e_2), \quad (48)$$

where $\mathcal{C} = \text{diag}(c_i)$ with $c_i > 0$ for $i = 1, \dots, n$, and p and q (where $p > q$) are positive odd numbers.

The control torque obtained from the output of the TSMC is expressed as follows:

$$\tau = \bar{M}(\alpha_1) \left(\dot{\alpha}_{2d} + \mathcal{C}^{\frac{q}{p}} e_2^{2-\frac{p}{q}} + \Lambda \text{sign}(S) + \Delta S \right) + \bar{C}(\alpha_1, \alpha_2)\alpha_2 + \bar{G}(\alpha_1) \quad (49)$$

where $\Delta = \text{diag}(\delta_i)$ with $\delta_i > 0$ for $i = 1, \dots, n$, and $|\mathcal{F}| \leq \Lambda$.

In the FTSMC approach [47], a sliding function is constructed as follows:

$$S = e_2 + \mathcal{C}\text{Sig}^\zeta(e_1) + \omega\text{Sig}^\beta(e_1) \quad (50)$$

where $\mathcal{C} = \text{diag}(c_i)$ with $c_i > 0$, $\omega = \text{diag}(\omega_i)$ with $\omega_i > 0$, ζ and β are parameter vectors with elements $\zeta_i > 1$ and $0 < \beta_i < 1$, respectively, for $i = 1, \dots, n$.

The control torque obtained from the output of the FTSMC is expressed as follows:

$$\tau = \bar{M}(\alpha_1)(\dot{\alpha}_{2d} + \mathcal{Z} + \Lambda\text{sign}(S) + \Delta S) + \bar{C}(\alpha_1, \alpha_2)\alpha_2 + \bar{G}(\alpha_1) \quad (51)$$

where $\Delta = \text{diag}(\delta_i)$ with $\delta_i > 0$, $\mathcal{Z} = [z_1, \dots, z_n]$ with $z_i = \zeta_i c_i |e_i|^{\zeta_i-1} \dot{e}_i + \beta_i \omega_i |e_i|^{\beta_i-1} \dot{e}_i$, for $i = 1, \dots, n$, and $|\mathcal{F}| \leq \Lambda$.

Assuming we do not have knowledge of all the robot dynamics, including $\bar{M}(\alpha_1)$, $\bar{C}(\alpha_1, \alpha_2)$, $\bar{G}(\alpha_1)$, and the uncertain term $\mathcal{F}$, we have two options for controlling the robot: the PID controller or the developed algorithm.

The control torque obtained from the output of the PID [5] is expressed as follows:

$$\tau = -K_P e_1 - K_I \int e_1 - K_D e_2 \quad (52)$$

where the control gains K_P , K_I , and K_D are positive.

5.2. Simulation Results and Discussion: Exploring Findings and Analysis

The tracking accuracy is assessed using the root-mean-square (RMS) algorithm, and the resulting RMS errors are provided in Table 4. Each joint's RMS error is denoted as ϵ_1 (Joint 1), ϵ_2 (Joint 2), and ϵ_3 (Joint 3).

Table 4. Comparison of the control performance from five controllers for trajectory tracking.

PID ($\epsilon_{1,2,3}$)	SMC ($\epsilon_{1,2,3}$)	TSMC ($\epsilon_{1,2,3}$)	FTSMC ($\epsilon_{1,2,3}$)	Developed Algorithm ($\epsilon_{1,2,3}$)
1.4×10^{-3}	1.890×10^{-4}	2.475×10^{-5}	2.383×10^{-5}	7.2×10^{-7}
3.45×10^{-2}	3.376×10^{-4}	4.672×10^{-5}	3.941×10^{-5}	1.539×10^{-5}
7.4×10^{-3}	3.502×10^{-4}	5.264×10^{-5}	5.030×10^{-5}	1.64×10^{-6}

Figure 3 demonstrates the performance of an RBFNN in approximating the dynamics of the robot and handling unknown uncertain terms. It is evident that by combining the benefits of RBFNN and PPCA, the system can quickly obtain accurate approximations for the entire robot dynamics and unknown uncertain terms.

In Figure 4, the robot's end-effector trajectory is manipulated to closely track the desired path provided in Table 2, aiming for the highest possible accuracy. The model-based methods, including SMC, TSMC, and FTSMC, exhibit matching with the desired trajectory, showcasing their effectiveness. Additionally, the proposed model-free method also demonstrates close adherence to the desired trajectory. In contrast, the trajectory produced by the PID controller exhibits a noticeable deviation from the desired path, indicating a relatively lower level of accuracy.

To thoroughly analyze and evaluate the effectiveness of each method, we rely on the simulation results presented in Figures 5–8 and Table 4. These results allow for a precise and specific examination of the performance of each method, providing valuable insights into their respective strengths and limitations.

Upon closer examination of the smaller figures within Figures 5–7, which depict a zoomed-in view of the phase from 0 to 1 second, it becomes evident that only the developed method exhibits convergence within the prescribed performance. This indicates that the developed method achieves fixed-time convergence, whereas the other methods do not demonstrate the same level of convergence.

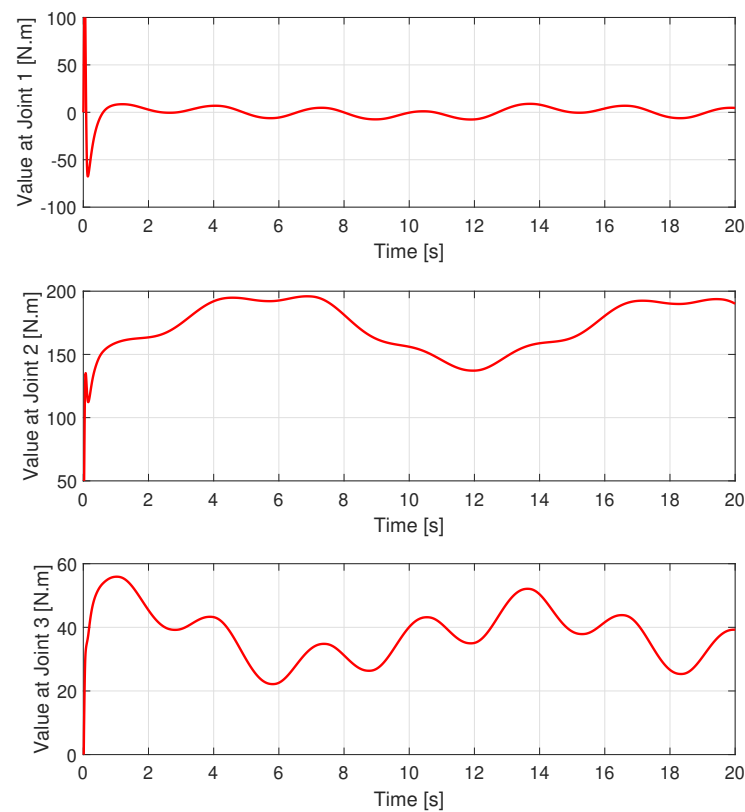


Figure 3. Performance of an RBFNN in approximating robot dynamics and unknown uncertain terms.

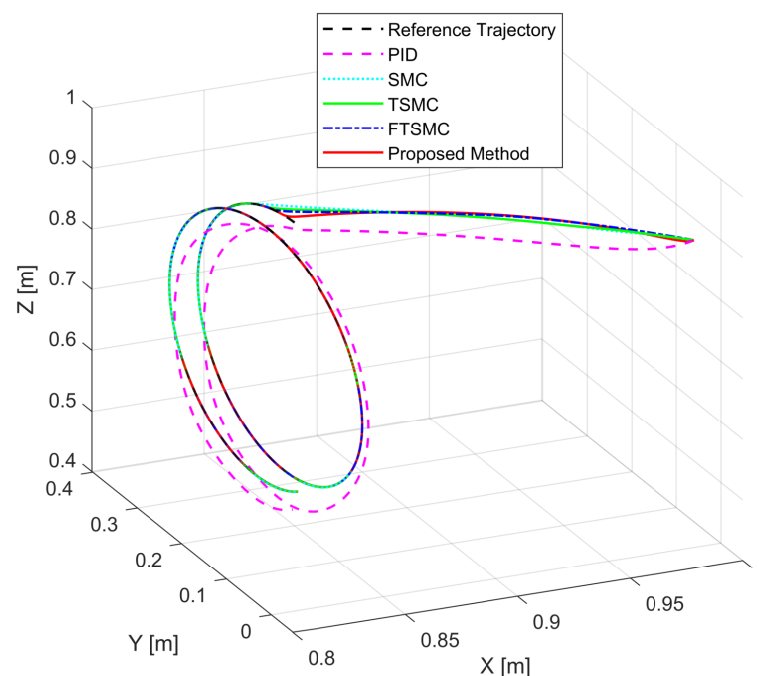


Figure 4. Analyzing the effects of different controllers on the end-effector trajectory of the robot.

Furthermore, when analyzing the smaller figures within Figures 5–7, which depict a zoomed-in view of the phase from the 2nd to 20th second, it becomes apparent that the PID controller fails to deliver satisfactory tracking performance during normal operations. This can be attributed to its inability to handle uncertainties and disturbances, which are

commonly present in the system. The PID controller exhibits large tracking errors in the presence of uncertainties and disturbances.

On the other hand, the SMC, TSMC, and FTSMC controllers demonstrate improved tracking performances with steady-state errors within predefined bounds. Although the SMC controller handles uncertainties and disturbances better than the PID controller, its performance still falls short of the desired level when compared to the TSMC and FTSMC controllers.

In contrast, despite the influence of unknown robot dynamics and uncertain terms, the proposed controller, designed appropriately, yields the smallest steady-state errors among all the methods considered.

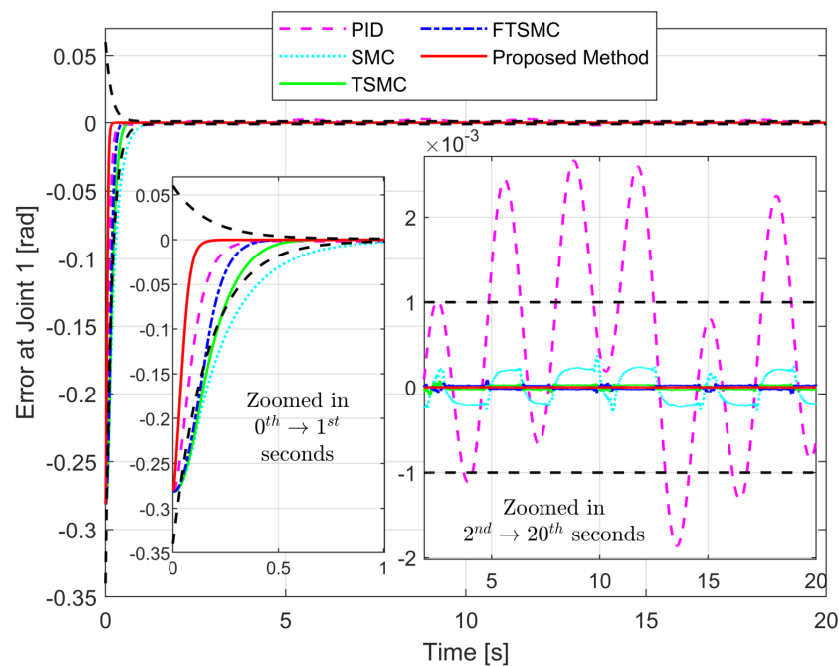


Figure 5. Analyzing the temporal variation of the tracking error in the first joint of a robot manipulator.

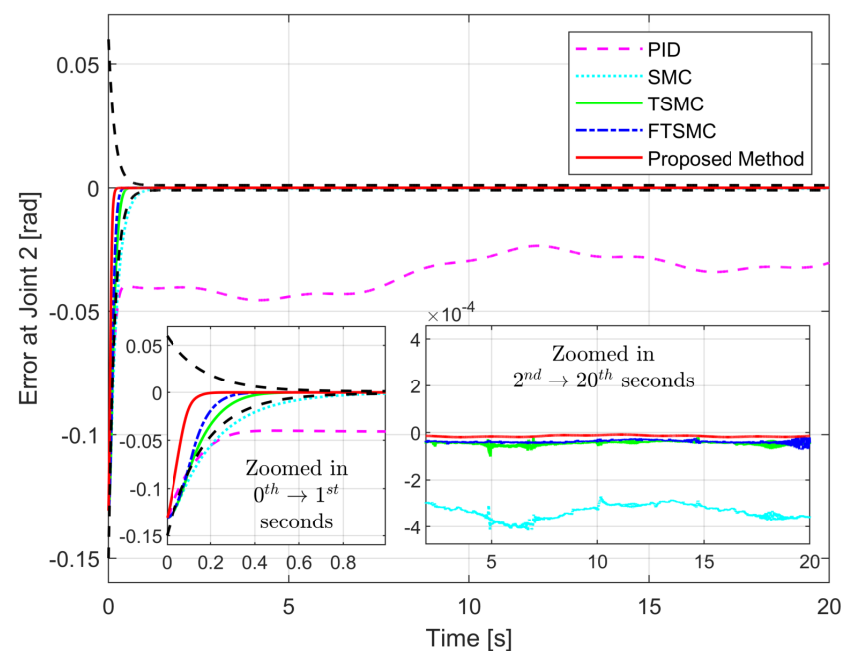


Figure 6. Analyzing temporal variation of the tracking error in the second joint of a robot manipulator.

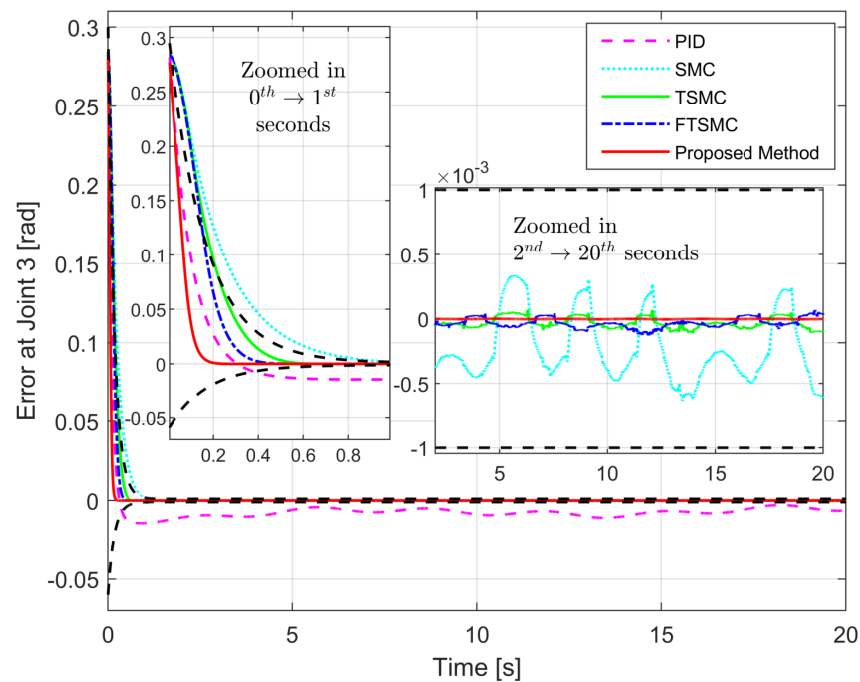


Figure 7. Analyzing temporal variation of the tracking error in the third joint of a robot manipulator.

As a result of tracking in the joint space, it is observed that the proposed controller achieves the smallest tracking errors, indicating the highest tracking accuracy along the XYZ axes in three dimensions, as depicted in Figure 8. This signifies that the proposed controller successfully minimizes the deviations between the desired and actual trajectories, resulting in superior tracking performance characterized by enhanced accuracy and precision. It is worth noting that the other controllers, namely SMC, TSMC, and FTSMC, exhibit similar trends, showcasing their respective effectiveness in achieving accurate tracking as well. However, it should be mentioned that the PID controller shows poorer tracking performance compared to the other methods, as evident from the larger tracking errors in the joint space.

In Figure 9, the control inputs for each controller are displayed. The proposed controller exhibits smooth and comparable control efforts to the PID controller while delivering superior tracking performance. On the other hand, controllers like SMC, TSMC, and FTSMC demonstrate good tracking performance, but their control inputs exhibit chattering behavior. Based on this simulation study, it can be concluded that the proposed controller is more effective in achieving tracking control for a robot in the presence of uncertain terms and unknown dynamics compared to the other controllers. The proposed controller strikes a balance between smooth control and accurate tracking, making it a favorable choice for practical applications.

Based on the simulation results, it is evident that the proposed controller surpasses other controllers in terms of trajectory tracking performance, demonstrating reduced settling time and overshoot. Additionally, it exhibits improved robustness against unknown dynamics, external disturbances, and uncertainties. Moreover, the chattering effect, commonly observed in sliding mode-based controllers such as SMC, TSMC, and FTSMC, is significantly mitigated by incorporating an NN for the real-time approximation of unknown dynamics, resulting in a smoother control signal. Overall, the simulation results validate the effectiveness and superiority of the proposed controller, specifically the TSMC design framework based on an RBFNN and PPCA, in achieving precise trajectory tracking performance with enhanced robustness and minimized chattering effect.

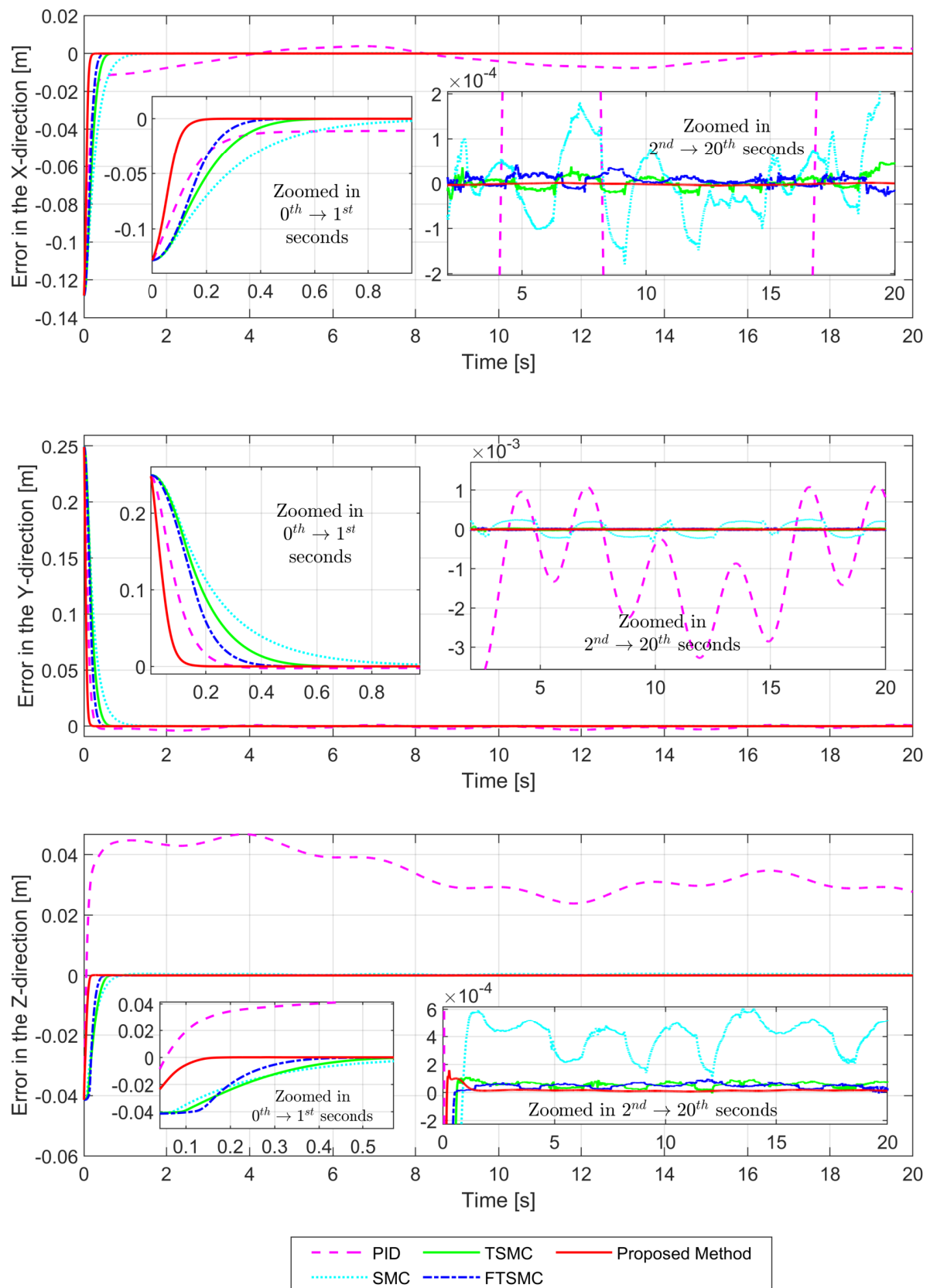


Figure 8. Examining the time evolution of the X, Y, and Z axis errors in the desired and actual trajectories.

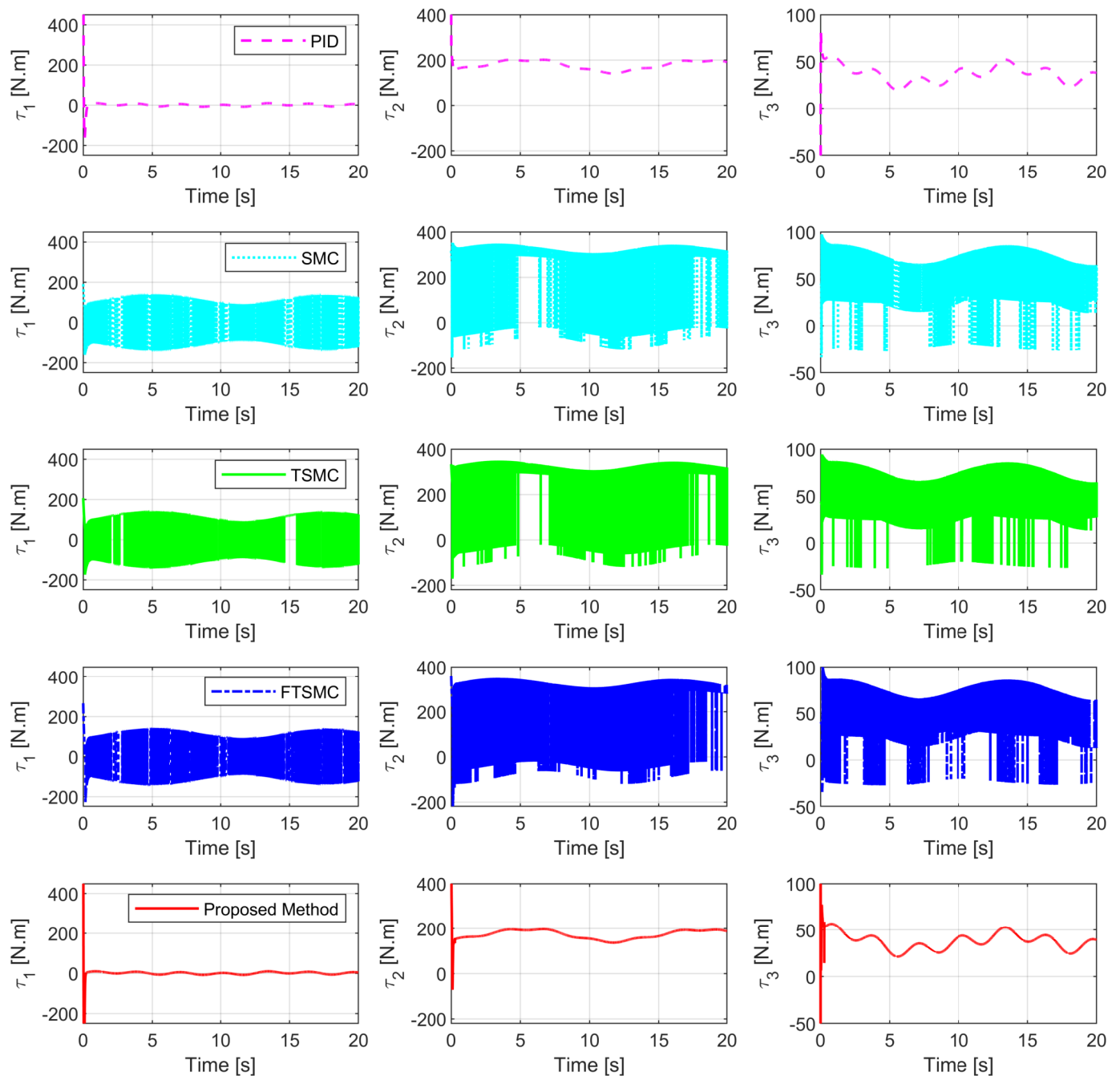


Figure 9. Investigating input control signals generated by control methods.

6. Conclusions

In conclusion, this paper proposes a novel approach to address three significant challenges in controlling robot manipulators, namely response time improvement, tracking error and chattering minimization, and controller robustness enhancement. The proposed approach uses a fixed-time sliding mode function that employs transformation errors to achieve the prescribed control performance while keeping maximum overshoot, convergence time, and tracking errors within predefined bounds. Additionally, the use of an RBFNN eliminates the need for knowledge of the robot's dynamical properties and uncertain terms, reducing negative chattering. The novel fixed-time TSMC algorithm for robot manipulators does not require their dynamical model and has been shown to ensure global fixed-time convergence for both tracking errors and adaptive laws using the Lyapunov

theory. The proposed method's effectiveness has been demonstrated through simulations on a robot manipulator.

Our future research direction will be to extend the application of the proposed controller to a practical system known as the 6-DOF FARA SAMSUNG Robot Manipulator as well as other classes of second-order nonlinear systems.

Author Contributions: Methodology, conceptualization, writing—original draft preparation, and writing—review and editing, A.T.V.; validation, software, visualization, and resources, T.N.T.; supervision, funding acquisition, and project administration, H.-J.K.; formal analysis, investigation, and data curation, A.T.V. and H.-J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University of Ulsan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The corresponding author can offer the datasets generated and/or analyzed during our study upon reasonable request.

Acknowledgments: This research was supported by the University of Ulsan.

Conflicts of Interest: The authors declare that there are no conflict of interest.

References

1. Sciavicco, L.; Siciliano, B. *Modelling and Control of Robot Manipulators*; Springer: Berlin/Heidelberg, Germany, 2001.
2. Sanchez, J.; Corrales, J.A.; Bouzgarrou, B.C.; Mezouar, Y. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey. *Int. J. Robot. Res.* **2018**, *37*, 688–716.
3. Rout, A.; Deepak, B.; Biswal, B.B.; Mahanta, G.B. Weld seam detection, finding, and setting of process parameters for varying weld gap by the utilization of laser and vision sensor in robotic arc welding. *IEEE Trans. Ind. Electron.* **2021**, *69*, 622–632. [\[CrossRef\]](#)
4. Markus, E.D.; Agee, J.T.; Jimoh, A.A. Flat control of industrial robotic manipulators. *Robot. Auton. Syst.* **2017**, *87*, 226–236. [\[CrossRef\]](#)
5. Rocco, P. Stability of PID control for industrial robot arms. *IEEE Trans. Robot. Autom.* **1996**, *12*, 606–614. [\[CrossRef\]](#)
6. Llama, M.A.; Kelly, R.; Santibañez, V. Stable computed-torque control of robot manipulators via fuzzy self-tuning. *IEEE Trans. Syst. Man Cybern. Part (Cybern.)* **2000**, *30*, 143–150. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Peng, W.; Lin, Z.; Su, J. Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques. *IET Control. Theory Appl.* **2009**, *3*, 701–711. [\[CrossRef\]](#)
8. Van, M.; Do, X.P.; Mavrouniotis, M. Self-tuning fuzzy PID-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. *ISA Trans.* **2020**, *96*, 60–68. [\[CrossRef\]](#)
9. Cao, Y.; Song, Y.D. Adaptive PID-like fault-tolerant control for robot manipulators with given performance specifications. *Int. J. Control.* **2020**, *93*, 377–386. [\[CrossRef\]](#)
10. Le, T.D.; Kang, H.J.; Suh, Y.S.; Ro, Y.S. An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator. *Neurocomputing* **2013**, *116*, 53–61. [\[CrossRef\]](#)
11. Patiño, H.D.; Carelli, R.; Kuchen, B.R. Neural networks for advanced control of robot manipulators. *IEEE Trans. Neural Netw.* **2002**, *13*, 343–354. [\[CrossRef\]](#)
12. Homayounzade, M.; Khademhosseini, A. Disturbance observer-based trajectory following control of robot manipulators. *Int. J. Control. Autom. Syst.* **2019**, *17*, 203–211. [\[CrossRef\]](#)
13. Vo, A.T.; Truong, T.N.; Kang, H.J. Fixed-Time RBFNN-Based Prescribed Performance Control for Robot Manipulators: Achieving Global Convergence and Control Performance Improvement. *Mathematics* **2023**, *11*, 2307. [\[CrossRef\]](#)
14. Vo, A.T.; Truong, T.N.; Kang, H.J. A Novel Tracking Control Algorithm With Finite-Time Disturbance Observer for a Class of Second-Order Nonlinear Systems and its Applications. *IEEE Access* **2021**, *9*, 31373–31389. [\[CrossRef\]](#)
15. Edwards, C.; Colet, E.F.; Fridman, L.; Colet, E.F.; Fridman, L.M. *Advances in Variable Structure and Sliding Mode Control*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 334.
16. Utkin, V.I.; Vadim, I. Sliding mode control. In *Variable Structure Systems: From Principles to Implementation*; IET: Stevenage, UK, 2004; Volume 66, p. 1.
17. Li, R.; Yang, L.; Chen, Y.; Lai, G. Adaptive sliding mode control of robot manipulators with system failures. *Mathematics* **2022**, *10*, 339. [\[CrossRef\]](#)

18. Ferrara, A.; Incremona, G.P. Design of an integral suboptimal second-order sliding mode controller for the robust motion control of robot manipulators. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 2316–2325. [\[CrossRef\]](#)
19. Nguyen, V.C.; Vo, A.T.; Kang, H.J. A finite-time fault-tolerant control using non-singular fast terminal sliding mode control and third-order sliding mode observer for robotic manipulators. *IEEE Access* **2021**, *9*, 31225–31235. [\[CrossRef\]](#)
20. Vo, A.T.; Kang, H.J. Adaptive neural integral full-order terminal sliding mode control for an uncertain nonlinear system. *IEEE Access* **2019**, *7*, 42238–42246. [\[CrossRef\]](#)
21. Lee, J.; Chang, P.H.; Jin, M. Adaptive integral sliding mode control with time-delay estimation for robot manipulators. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6796–6804. [\[CrossRef\]](#)
22. Vo, A.T.; Kang, H.J. Neural Integral Non-Singular Fast Terminal Synchronous Sliding Mode Control for Uncertain 3-DOF Parallel Robotic Manipulators. *IEEE Access* **2020**, *8*, 65383–65394. [\[CrossRef\]](#)
23. Ning, B.; Han, Q.L.; Lu, Q. Fixed-time leader-following consensus for multiple wheeled mobile robots. *IEEE Trans. Cybern.* **2019**, *50*, 4381–4392. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Vo, A.T.; Truong, T.N.; Kang, H.J.; Van, M. A Robust Observer-Based Control Strategy for n-DOF Uncertain Robot Manipulators with Fixed-Time Stability. *Sensors* **2021**, *21*, 7084. [\[CrossRef\]](#)
25. Sai, H.; Xu, Z.; Xia, C.; Sun, X. Approximate continuous fixed-time terminal sliding mode control with prescribed performance for uncertain robotic manipulators. *Nonlinear Dyn.* **2022**, *110*, 431–448. [\[CrossRef\]](#)
26. Anjum, Z.; Zhou, H.; Ahmed, S.; Guo, Y. Fixed time sliding mode control for disturbed robotic manipulator. *J. Vib. Control* **2023**, 10775463231165094. [\[CrossRef\]](#)
27. Zhang, L.; Wang, Y.; Hou, Y.; Li, H. Fixed-time sliding mode control for uncertain robot manipulators. *IEEE Access* **2019**, *7*, 149750–149763. [\[CrossRef\]](#)
28. Xie, Y.; Ma, Q.; Gu, J.; Zhou, G. Event-triggered fixed-time practical tracking control for flexible-joint robot. *IEEE Trans. Fuzzy Syst.* **2022**, *31*, 67–76. [\[CrossRef\]](#)
29. Van, M.; Sun, Y.; McIlvanna, S.; Nguyen, M.N.; Zocco, F.; Liu, Z.; Wang, H.C. Distributed Fixed-Time Consensus Control for Multiple AUV Systems with Input Saturations. *arXiv* **2023**, arXiv:2302.14162.
30. Zhang, L.; Liu, J.; Cui, N. Backstepping control for a two-link manipulator with appointed-time convergence. *ISA Trans.* **2022**, *128*, 208–219. [\[CrossRef\]](#)
31. Van, M.; Sun, Y.; McIlvanna, S.; Nguyen, M.N.; Khyam, M.O.; Ceglarek, D. Adaptive Fuzzy Fault Tolerant Control for Robot Manipulators with Fixed-Time Convergence. *IEEE Trans. Fuzzy Syst.* **2023**. [\[CrossRef\]](#)
32. Song, Z.; Sun, K. Prescribed performance adaptive control for an uncertain robotic manipulator with input compensation updating law. *J. Frankl. Inst.* **2021**, *358*, 8396–8418. [\[CrossRef\]](#)
33. Liu, Z.; Zhao, Y.; Zhang, O.; Chen, W.; Wang, J.; Gao, Y.; Liu, J. A Novel Faster Fixed-Time Adaptive Control for Robotic Systems With Input Saturation. *IEEE Trans. Ind. Electron.* **2023**. [\[CrossRef\]](#)
34. Hu, Y.; Yan, H.; Zhang, H.; Wang, M.; Zeng, L. Robust adaptive fixed-time sliding-mode control for uncertain robotic systems with input saturation. *IEEE Trans. Cybern.* **2022**, *53*, 2636–2646. [\[CrossRef\]](#)
35. Kavitha, A.; Reddy, V.B.; Singh, N.; Gunjan, V.K.; Lakshmana, K.; Khan, A.A.; Wechtaisong, C. Security in IoT Mesh Networks Based on Trust Similarity. *IEEE Access* **2022**, *10*, 121712–121724. [\[CrossRef\]](#)
36. Truong, T.N.; Vo, A.T.; Kang, H.J. An Adaptive Terminal Sliding Mode Control Scheme via Neural Network Approach for Path-following Control of Uncertain Nonlinear Systems. *Int. J. Control Autom. Syst.* **2022**, *20*, 2081–2096. [\[CrossRef\]](#)
37. Sun, Y.; Van, M.; McIlvanna, S.; Nhat, N.M.; McLoone, S.; Ceglarek, D.; Ge, S.S. Fixed-time Adaptive Neural Control for Physical Human-Robot Collaboration with Time-Varying Workspace Constraints. *arXiv* **2023**, arXiv:2303.02456.
38. Fang, H.; Wu, Y.; Xu, T.; Wan, F. Adaptive neural sliding mode control of uncertain robotic manipulators with predefined time convergence. *Int. J. Robust Nonlinear Control* **2022**, *32*, 9213–9238. [\[CrossRef\]](#)
39. Wu, Y.; Fang, H.; Xu, T.; Wan, F. Adaptive neural fixed-time sliding mode control of uncertain robotic manipulators with input saturation and prescribed constraints. *Neural Process. Lett.* **2022**, *54*, 3829–3849. [\[CrossRef\]](#)
40. Bechlioulis, C.P.; Rovithakis, G.A. Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance. *IEEE Trans. Autom. Control* **2008**, *53*, 2090–2099. [\[CrossRef\]](#)
41. Truong, T.N.; Vo, A.T.; Kang, H.J. Real-Time Implementation of the Prescribed Performance Tracking Control for Magnetic Levitation Systems. *Sensors* **2022**, *22*, 9132. [\[CrossRef\]](#)
42. Tran, X.T.; Oh, H. Prescribed performance adaptive finite-time control for uncertain horizontal platform systems. *ISA Trans.* **2020**, *103*, 122–130. [\[CrossRef\]](#)
43. Zhou, Z.G.; Zhou, D.; Shi, X.N.; Li, R.F.; Kan, B.Q. Prescribed performance fixed-time tracking control for a class of second-order nonlinear systems with disturbances and actuator saturation. *Int. J. Control* **2021**, *94*, 223–234. [\[CrossRef\]](#)
44. Polyakov, A. Nonlinear feedback design for fixed-time stabilization of linear control systems. *IEEE Trans. Autom. Control* **2011**, *57*, 2106–2110. [\[CrossRef\]](#)
45. Zuo, Z. Nonsingular fixed-time consensus tracking for second-order multi-agent networks. *Automatica* **2015**, *54*, 305–309. [\[CrossRef\]](#)

46. Yu, S.; Yu, X.; Shirinzadeh, B.; Man, Z. Continuous finite-time control for robotic manipulators with terminal sliding mode. *Automatica* **2005**, *41*, 1957–1964. [[CrossRef](#)]
47. Bhat, S.P.; Bernstein, D.S. Finite-time stability of continuous autonomous systems. *SIAM J. Control. Optim.* **2000**, *38*, 751–766. [[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.