

Article

# Adaptive Fault-Tolerant Tracking Control for Multi-Joint Robot Manipulators via Neural Network-Based Synchronization

Quang Dan Le  and Erfu Yang \* 

Robotics and Autonomous Systems Group, Department of Design, Manufacturing and Engineering Management, University of Strathclyde, Glasgow G1 1XJ, UK; l.quang-dan@strath.ac.uk

\* Correspondence: erfu.yang@strath.ac.uk

**Abstract:** In this paper, adaptive fault-tolerant control for multi-joint robot manipulators is proposed through the combination of synchronous techniques and neural networks. By using a synchronization technique, the position error at each joint simultaneously approaches zero during convergence due to the constraints imposed by the synchronization controller. This aspect is particularly important in fault-tolerant control, as it enables the robot to rapidly and effectively reduce the impact of faults, ensuring the performance of the robot when faults occur. Additionally, the neural network technique is used to compensate for uncertainty, disturbances, and faults in the system via online updating. Firstly, novel robust synchronous control for a robot manipulator based on terminal sliding mode control is presented. Subsequently, a combination of the novel synchronous control and neural network is proposed to enhance the fault tolerance of the robot manipulator. Finally, simulation results for a 3-DOF robot manipulator are presented to demonstrate the effectiveness of the proposed controller in comparison to traditional control techniques.

**Keywords:** fault-tolerant control; passive fault-tolerant control; robot manipulator; adaptive control; synchronization; neural network



**Citation:** Le, Q.D.; Yang, E. Adaptive Fault-Tolerant Tracking Control for Multi-Joint Robot Manipulators via Neural Network-Based Synchronization. *Sensors* **2024**, *24*, 6837. <https://doi.org/10.3390/s24216837>

Academic Editor: Carlo Alberto Avizzano

Received: 15 September 2024

Revised: 7 October 2024

Accepted: 21 October 2024

Published: 24 October 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

During the past few decades, robots have played and continue to play an important role in many areas, especially in the advancement of various industries and replacement of humans in harsh and hazardous work environments. Consequently, they have been increasingly studied and developed to enhance their working capabilities and safety in many circumstances and diverse work environments. In robot manipulator systems, the occurrence of faults in actuators or sensors can reduce the performance of the robot, damage the system, and have economic implications [1,2]. Therefore, fault-tolerant control (FTC) is introduced to improve the safety of robot systems. It guarantees the acceptable performance and provides operators with a scheduled maintenance time to ensure smooth operation of the robotics system. In general, fault-tolerant control can be divided into two types: (1) active fault-tolerant control (AFTC) [3–5] and (2) passive fault-tolerant control (PFTC) [6,7].

In AFTC, to ensure the stability of the system, fault detection and fault diagnosis (FDD) are used to compensate with a conventional controller. The performance of AFTC highly depends on the accuracy of the FDD process. Therefore, numerous FDD methodologies have been developed to enhance the accuracy of FDD [8–10]. In robot manipulator systems, the FDD is typically supplanted by fault estimation (FE), a process that comprehensively estimates, detects, and isolates faults in a single step. This AFTC structure control is similar to disturbance observer-based (DOB) control approaches [11–13]. The key distinction between AFTC and DOB lies in the utilization of the estimated value from FE, which is compared with a predefined threshold for fault detection and isolation. Additionally, these estimated values serve to alert operators, prompting them to take efficient actions for

maintenance. A notable advancement of AFTC is the ability to deal with high-magnitude faults or multi-fault scenarios. However, AFTC has a slow response to faults due to the delayed feedback from FE. The consequential effect is the occurrence of peaking phenomena, which reduce the performance of the robot system [3,14].

In contrast, PFTC uses a single controller for both normal and fault condition. This structural control approach enables the robot to respond more rapidly to faults than the AFTC scheme. Therefore, peaking phenomena are reduced in comparison with the AFTC scheme. Sliding Mode Control (SMC) stands out as a popular technique considered for PFTC. Given its robustness and ability to deal with disturbances and uncertainties, SMC has received significant attention in PFTC research [15–17]. In addition, the finite and fixed-time convergence are advantageous features of SMC, which have attracted the efforts of many researchers [18–20]. However, relying solely on the SMC technique renders the robot system less effective in addressing high-magnitude faults. The second most popular PFTC approach is adaptive control [21,22], which can handle high-magnitude faults. It quickly adjusts the control strategy in response to faults, with the capability of real-time adjustments. In addition, this technique can deliver high performance, even with less-precise dynamic robot models. Based on the analysis above, it can be seen that the choice between AFTC and PFTC depends on the model of the system and the understanding the system designer, with the potential to optimize the performance of the robot system. This paper focuses on the development of PFTC to provide a rapid response to faults and enhance the system's ability to handle high-magnitude faults based on an adaptive technique.

To enhance the response of a robotic manipulator system to faults, the synchronization technique was used [3,18,23]. The synchronization technique was introduced in [24,25]. This method was well known in closed-loop robot systems like the dual-drive gantry mechanism [26], parallel robot manipulator [27,28], cable-driven robot manipulator [29], multi-robot cooperation [30], etc. The synchronization technique based on the cross-error was developed by Y. Koren [24] for computerized numerical control (CNC) machines. Its main idea is that dependence axes can be controlled by considering the effects of the other axes through cross-error. In general, in closed-loop mechanism systems, internal tensor force occurs during the motion of the actuator. This force can degrade the performance or damage the system. The synchronous technique causes the position error of each joint to simultaneously approach zero; therefore, it reduces the internal tensor force. In contrast, serial robots with open-loop mechanisms may not contain the internal tensor force due to the consideration of the dynamic model. Therefore, the synchronization will not provide an advantage when it is compared with model-base controllers in open-loop systems. However, the feature of each position error simultaneously approaching zero still exists when the synchronization control is applied to a serial robot. This feature will help the robot system rapidly respond to faults. Therefore, the performance of the robot will increase when faults occur in the system. Synchronous control techniques applied to fault-tolerant control of serial manipulators were first introduced in [3,18,23]. Experimental results on real robotic systems have demonstrated the unique advantages of this technique when dealing with high-magnitude faults. Despite the proven effectiveness of synchronous control, the authors have also pointed out limitations when combined with an extended state observer in AFTC schemes. Specifically, the peaking phenomenon arises when faults initially occur in the system. This happens due to the delayed response of the controller, which relies on fault information from the fault estimator. This phenomenon significantly reduces the system's performance. Unlike the approaches in [3,18,23], this paper utilizes synchronization within the PFTC scheme to enhance the response, reduce the peaking phenomena, and improve the ability to handle high-magnitude faults. A novel synchronization control is proposed. The stability of the system is proved by Lyapunov theory.

The adaptive methodology based on intelligence techniques has received significant attention from researchers and engineers across various domains in automatic control. This approach may rely on either fuzzy logic techniques [31,32] or neural networks [33–35]. Fuzzy logic control typically requires the experience of an operator to design the fuzzy rules,

while neural networks eliminate the operator's need for specific knowledge of network design. Two types of popular neural network architectures in controllers are feed-forward networks and recurrent networks. However, these kinds of neural networks have some drawbacks such as local minima, slow convergence, and challenges in selecting learning rate and initial weight parameters. To address these limitations, researchers have proposed various solutions, as seen in [36,37], where the authors provided a technique for selecting initial weights and improving the convergence time. Another strategy to enhance the convergence time and overcome the local minimum is the use of a momentum part, as suggested in [38,39]. In addition, orthogonal neural networks (ONNs) were presented by Yen and Chen in [40] to overcome the drawbacks of traditional neural network architectures. This kind of neural network has been applied to deep neural networks [41–43] and shown to be effective in comparison to feed-forward networks. In automatic control, ONNs have been applied to anti-lock braking systems [44] and robot control [28,45]. In this paper, the ONN is designed to increase the ability to manage high-magnitude faults and provide fast responses to faults based on the online compensation of the ONN.

In this paper, an adaptive fault-tolerant control for multi-joint robot manipulators based on a combination of synchronous techniques and neural networks is proposed to enhance performance. Initially, the PFTC synchronous terminal sliding mode controller (PFTC S-TSMC) is investigated by exploiting the synchronization error with a novel synchronous sliding surface. Subsequently, an adaptive fault-tolerant control based on the combination of synchronization controller and ONN (PFTC NN-S-TSMC) is put forward to improve the ability to manage high-magnitude faults. Finally, both proposed PFTC controllers are verified on a 3-DOF robot manipulator using MATLAB Simulink to demonstrate the advantages of the proposed controller. The contributions of this paper can be summarized as follows:

- (1) A passive fault-tolerant control scheme based on a novel robust synchronous terminal sliding mode control is designed. By simultaneously driving each joint position error to zero using a synchronization technique, the system exhibits a rapid response to faults. This approach mitigates the peaking phenomenon typically seen in active fault-tolerant control methods, which often suffer from slow response times [18]. The proposed controller demonstrates a faster response to faults and is more effective in handling high-magnitude faults compared to conventional terminal sliding mode control techniques [46]. The enhanced performance of the fault-tolerant control is attributed to the incorporation of additional joint information, which further improves the system's effectiveness.

- (2) An adaptive fault-tolerant control based on the neural network synchronous terminal sliding mode is proposed. Due to the combination of the advantages of the synchronization technique and online adjustment of the neural network, the proposed controller outperforms the robust proposed controller and conventional terminal sliding mode controller [46]. By using the orthogonal neural network, the advantages in low-level control systems are shown. Unlike in [47], using the position error, the input to the orthogonal neural network in the proposed controller is the synchronization error, ensuring that the synchronization feature is maintained during the neural network's online weight updates.

- (3) The stability of the system is proven with Lyapunov theory for both of the proposed controllers.

The rest of this paper is organized as follows. In Section 2, the dynamic model of the robot, faults, and orthogonal neural network are given. The novel synchronization sliding surface, proposed robust synchronous terminal sliding mode control, and an adaptive fault-tolerant control based on synchronization and orthogonal neural networks are proposed in Section 3. In Section 4, a simulation of the proposed control in a 3-DOF robot manipulator using MATLAB Simulink is presented. A discussion of the limitations and future work is provided in Section 5. Finally, the conclusions are given in Section 6.

## 2. Preliminaries

### 2.1. Dynamic Model of Robot Manipulators

The dynamic model of an  $n$ -degree robot manipulator is governed by the following equation [48]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \Xi = \tau \quad (1)$$

where  $\ddot{q}, \dot{q}, q \in \mathbb{R}^n$  are the vectors of joint acceleration, velocity, and position, respectively.  $M(q) \in \mathbb{R}^{n \times n}$ ,  $C(q, \dot{q}) \in \mathbb{R}^n$ , and  $G(q) \in \mathbb{R}^n$  represent the inertia matrix, the centripetal and Coriolis matrix, and the gravitation force, respectively.  $\tau \in \mathbb{R}^n$  is the torque at the joints.  $\Xi \in \mathbb{R}^n$  represents the unknown dynamic uncertainties and external disturbances.

**Assumption 1.** The matrices  $M(q)$  and  $\Xi$  are bounded by the following:

$$\begin{aligned} M_m &\leq \|M^{-1}(q)\| \leq M_M \\ \|\Xi\| &< a_0 + a_1\|q(t)\| + a_2\|\dot{q}(t)\|^2 + \gamma\|\tau\| \end{aligned}$$

where  $M_m, M_M$ , and  $\gamma$  are known as positive constants.

### 2.2. Model of Joint Actuator Faults

In robot manipulator systems, an actuator joint consists of many electrical and mechanical components such as an electronic motor, motor driver, gear, etc. Therefore, the occurrence of faults/failure at an actuator joint has a high probability. For example, a decrease in motor torque can result from increased friction of the stator and robot inside the electric motor after being in use for a long time, or it can result from increased friction of a mechanical part due to the environment such as temperature or dust. It can be considered as a loss-effectiveness fault. Generally, faults in robot manipulator systems can be represented as shown in Equation (2), where two common faults are loss-effectiveness and bias faults [49]:

$$\tau^t = (I - \rho(t))\tau + f(t) \quad (t > t_f) \quad (2)$$

where  $\tau^t \in \mathbb{R}^n$  is the vector of torques at the output joint actuator.  $\tau \in \mathbb{R}^n$  is the vector of torques at the output controller.  $\rho(t) = \text{diag}(\rho_i(t)) \in \mathbb{R}^{n \times n}$ ,  $0 \leq \rho_i(t) < 1$ ,  $i = (1, 2, \dots, n)$  denotes the loss-effective rate.  $f(t) \in \mathbb{R}^n$  is the vector of the bias fault.  $t_f$  is the time of the occurrence of faults.  $I \in \mathbb{R}^{n \times n}$  is an identity matrix.

Substituting Equation (2) into Equation (1), the dynamic model of an  $n$ -DOF robot manipulator with actuator faults can be written as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \Xi = (I - \rho(t))\tau + f(t) \quad (3)$$

In this paper, the investigation is focused on the effects of loss-effective and bias faults in robot manipulators. In loss-effective faults, when  $\rho_i(t) = 1$ , they are considered as lock-in-place faults, which are not investigated because the robot manipulator in this paper does not have a redundancy actuator. Therefore, the robot cannot tolerate this type of fault.

### 2.3. Orthogonal Neural Network

In this section, the conventional ONN is given. It is based on a feed-forward neural network (FNN) and polynomial functions as activity functions.

According to the theory of orthogonal function [50], an arbitrary function will have an orthogonal polynomial as follows:

$$F_n(x) = w_1\lambda_1(x) + w_2\lambda_2(x) + \dots + w_n\lambda_n(x) \quad (4)$$

such that

$$\lim_{n \rightarrow \infty} \int_a^b (f(x) - F_n(x))^2 dx = 0 \quad (5)$$

where

$$\int_a^b \lambda_i(x) \lambda_j(x) dx = \begin{cases} 0, & \text{if } i \neq j \\ A_j, & \text{if } i = j \end{cases} \quad (6)$$

$$w_i = \int_a^b f(x) \lambda_i(x) dx / A_i (i = 1, 2, \dots, n) \quad (7)$$

in which  $\{\lambda_1(x), \lambda_2(x), \dots\}$  is an orthogonal set. The orthogonal functions include Fourier series, the Bessel function, and Legendre and Chebyshev polynomials.

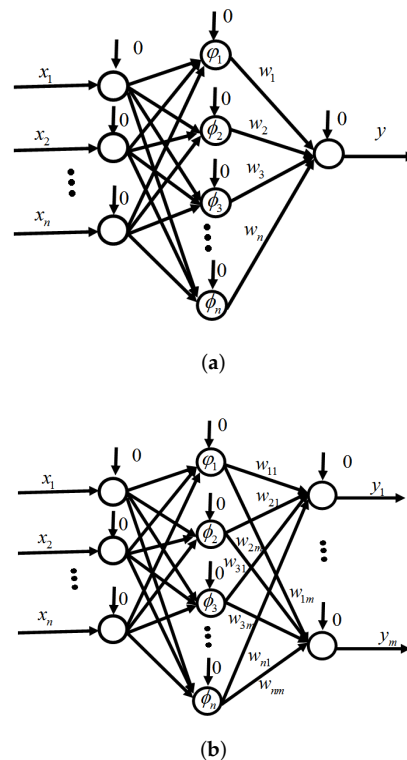
In the case where there is a function with  $m$  variables, an orthogonal function set is denoted as  $\{\Psi_1(x), \Psi_2(x), \dots\}$ . Then, each orthogonal function is defined as follows:

$$\Psi_i(X) = \lambda_{1i}(x_1) \lambda_{2i}(x_2) \dots \lambda_{mi}(x_n) \quad (8)$$

where  $X = [x_1, x_2, \dots, x_n]^T$  is the input vector.

An ONN based on a feed-forward network with one hidden layer is shown in Figure 1 by [40]. Figure 1 shows (a) multi input–one output and (b) multi input–multi output. The weight connections between the input layer and hidden layer are 1, and the bias is 0.  $\lambda_i (i = 0, 1, 2, \dots, m)$  is an orthogonal function. The weights between them are  $w_{ij}$  (where  $i$  is number of orthogonal functions and  $j$  is number of outputs). The output of the ONN [40] can be shown as follows:

$$y = F(X) = \sum_{i=1}^n w_i(t) \Psi_i(X) = W^T(t) \Psi \quad (9)$$



**Figure 1.** The orthogonal neural network. (a) Single output, (b) multiple outputs.

### 2.4. Definitions

Some definitions are necessary to propose the fault-tolerant control law.

**Definition 1.** We define  $[x]^\Lambda = [|x_1|^{\lambda_1} \text{sign}(x_1), |x_2|^{\lambda_2} \text{sign}(x_2), \dots, |x_n|^{\lambda_n} \text{sign}(x_n)] \in \mathfrak{R}^n$ , where  $\lambda_i > 0$  and  $\Lambda = \text{diag}(\lambda_i)$  with  $(i = 1, 2, \dots, n)$ .  $x = [x_1, x_2, \dots, x_n]^T \in \mathfrak{R}^n$ .  $y = [y_1, y_2, \dots, y_n]^T \in \mathfrak{R}^n$  and  $\text{sign}(x)$  is a signum function, which is defined as follows:

$$\text{sign}(x) = \begin{cases} -1, & \text{if } x < 0 \\ 0, & \text{if } x = 0 \\ 1, & \text{if } x > 0 \end{cases} \tag{10}$$

**Definition 2.** We define  $x \cdot y = [x_1 y_1, x_2 y_2, \dots, x_n y_n] \in \mathfrak{R}^n$ .

**Definition 3.** The time derivative of  $[x]^\Lambda$  is  $\frac{d}{dt}[x]^\Lambda = \Lambda|x|^{\Lambda-1} \cdot \dot{x} = [\lambda|x_1|^{\lambda-1} \dot{x}_1, \lambda|x_2|^{\lambda-2} \dot{x}_2, \dots, \lambda|x_n|^{\lambda-n} \dot{x}_n]^T$  and  $x^{\Lambda-1} = [x_1^{\lambda_1-1}, x_2^{\lambda_2-1}, \dots, x_n^{\lambda_n-1}] \in \mathfrak{R}^n$  where  $I = \text{diag}(1) \in \mathfrak{R}^{n \times n}$ .

### 3. Fault-Tolerant Control with Synchronous Terminal Sliding Mode Control

In this part, passive fault-tolerant control based on the synchronous terminal sliding mode control is proposed.

#### 3.1. Synchronization Error

A synchronization error [18] is given as follows:

$$\begin{cases} \varepsilon_1 = (1 + \psi_1 \psi_n) e_1 - \psi_1 e_2 - \psi_1 e_n \\ \varepsilon_2 = (1 + \psi_2 \psi_1) e_2 - \psi_2 e_3 - \psi_2 e_1 \\ \vdots \\ \varepsilon_n = (1 + \psi_n \psi_{n-1}) e_n - \psi_n e_2 - \psi_n e_1 \end{cases} \tag{11}$$

where  $e_i = q_{di} - q_i$  ( $i = 1, 2, \dots, n$ ) is the error at each joint, where  $q_{di}$  and  $q_i$  are the desired position and current position of the joint  $i$ , respectively.  $\psi_i (i = 1, 2, \dots, n)$  is the corresponding positive gain. In matrix form,

$$\varepsilon = T e \tag{12}$$

where  $\varepsilon = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]^T \in \mathfrak{R}^n$ ,  $e = [e_1, e_2, \dots, e_n]^T \in \mathfrak{R}^n$ ,  $T \in \mathfrak{R}^{n \times n}$  is a transformation matrix, and

$$T = \begin{bmatrix} (1 + \psi_1 \psi_n) & -\psi_2 & 0 & \dots & -\psi_n \\ -\psi_1 & (1 + \psi_2 \psi_1) & -\psi_3 & \dots & 0 \\ 0 & -\psi_2 & (1 + \psi_3 \psi_2) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\psi_1 & 0 & 0 & \dots & (1 + \psi_n \psi_{n-1}) \end{bmatrix} \tag{13}$$

#### 3.2. Proposed Robust Fault-Tolerant Control with Synchronization Error

In this section, robust fault-tolerant control with synchronization error based on terminal sliding mode control is proposed.

Equation (3) can be rewritten as follows:

$$\ddot{q} = M(q)^{-1}(\tau - H(q, \dot{q}) - \zeta) \tag{14}$$

where  $H(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q)$ ,  $\zeta = \rho(t)\tau + \Xi - f(t)$  represents the uncertainties, disturbances, and fault.

The proposed synchronous fast terminal sliding mode surface based on novel synchronous error is given as follows:

$$S = \dot{e} + \beta e + \alpha |\varepsilon|^\Lambda \quad (15)$$

where  $S = [s_1, s_2, s_2, \dots, s_n] \in \mathfrak{R}^n$  is the vector of the sliding surface.  $\alpha = \text{diag}(\alpha_i) \in \mathfrak{R}^{n \times n}$ ,  $\beta = \text{diag}(\beta_i) \in \mathfrak{R}^{n \times n}$ , and  $\Lambda = \text{diag}(\lambda_i) \in \mathfrak{R}^{n \times n}$  are the positive matrix gain, with  $0 \leq \lambda \leq 1$  and  $i = 1, 2, \dots, n$ .

From the novel synchronous sliding surface, the proposed passive fault-tolerant control with synchronous terminal sliding mode (PFTC S-TSMC) is given as follows:

$$\tau = \tau_0 + \tau_1 + \tau_2 \quad (16)$$

where

$$\tau_0 = M(q)\dot{q}_d + H(q, \dot{q}) \quad (17)$$

$$\tau_1 = M(q)(\alpha\Lambda|\varepsilon|^{\Lambda-1}\dot{e} + \beta\dot{e} + K\text{sign}(S)) \quad (18)$$

$$\tau_2 = \begin{cases} \frac{S}{\|S\|}w, & \|S\| \neq 0, \\ 0, & \|S\| = 0 \end{cases} \quad w = \frac{1}{1-\gamma}(a_0 + a_1\|q\| + a_2\|\dot{q}\|^2 + \gamma\|\tau_0 + \tau_1\|) \quad (19)$$

where  $\alpha = \text{diag}(\alpha_i) \in \mathfrak{R}_{n \times n}$ ,  $\beta = \text{diag}(\beta_i) \in \mathfrak{R}_{n \times n}$ ,  $K = \text{diag}(k_i) \in \mathfrak{R}_{n \times n}$ ,  $a_0, a_1, a_2$ , and  $\gamma$  are positive constants, and  $\gamma$  is given as follows:

$$\gamma = \frac{M_M - M_m}{M_M + M_m} \quad (20)$$

**Theorem 1.** *Given the uncertain robot manipulator system in Equation (3) and the proposed controller in Equation (16) and based on the novel synchronization error in Equation (12) and sliding surface in Equation (15), the tracking trajectory error of the robot manipulator converges to zero.*

**Proof of Theorem 1.** The Lyapunov function can be selected as follows:

$$V = \frac{1}{2}S^T S, \quad (21)$$

The time derivative of V in Equation (21) is as follows:

$$\begin{aligned} \dot{V} &= S^T \dot{S} \\ &= S^T (\ddot{e} + \alpha\Lambda|\varepsilon|^{\Lambda-1}\dot{e} + \beta\dot{e}) \\ &= S^T (\ddot{q}_d - \ddot{q} + \alpha\Lambda|\varepsilon|^{\Lambda-1}\dot{e} + \beta\dot{e}) \end{aligned} \quad (22)$$

Substituting Equation (14) into Equation (22) we have the following:

$$\dot{V} = S^T (\ddot{q}_d - (M(q)^{-1}(\tau - H(q, \dot{q}) - \zeta)) + \alpha\Lambda|\varepsilon|^{\Lambda-1}\dot{e} + \beta\dot{e}) \quad (23)$$

Substituting the proposed controller in Equation (16) into Equation (23), we obtain the following:

$$\dot{V} = -S^T M^{-1}(q) \left( \frac{S}{\|S\|} w - \zeta \right) - S^T K \text{sign}(S) \quad (24)$$

We can see that  $\|\tau\| \leq \|\tau_0 + \tau_1\| + \|\tau_2\|$ , and applying Assumption 1 yields the following:

$$\begin{aligned} \dot{V} &\leq -S^T M_m \left( \frac{S}{\|S\|} w - (a_0 + a_1 \|q(t)\| + a_2 \|\dot{q}(t)\|^2 + \gamma \|\tau\|) \right) - S^T K \text{sign}(S) \\ &\leq -S^T M_m \left( \frac{S}{\|S\|} w - (a_0 + a_1 \|q(t)\| + a_2 \|\dot{q}(t)\|^2 + \gamma \|\tau_0 + \tau_1\| + \gamma \|\tau_2\|) \right) - S^T K \text{sign}(S) \\ &= M_m (-\|S\| w + \|S\| (a_0 + a_1 \|q(t)\| + a_2 \|\dot{q}(t)\|^2 + \gamma \|\tau_0 + \tau_1\|) + \gamma \|\tau_2\|) - S^T K \text{sign}(S) \quad (25) \\ &= M_m \left( -\|S\| \frac{1}{1-\gamma} w + \|S\| \|w + \gamma \|\tau_2\| \|S\| \right) - S^T K \text{sign}(S) \\ &= M_m \left( \gamma \|\tau_2\| \|S\| - \frac{\gamma}{1-\gamma} \|S\| w \right) - S^T K \text{sign}(S) \end{aligned}$$

Substituting Equation (18) into Equation (25), we have the following:

$$\dot{V} \leq -S^T K \text{sign}(S) \leq -\sigma V^{\frac{1}{2}} \leq 0 \quad (26)$$

where  $\sigma = \lambda_{\min} K$ . When the sliding surface converges to zero, we have the position error  $e_i \rightarrow 0$ . Therefore, Theorem 1 is proven.  $\square$

### 3.3. Proposed Neural Network-Based Fault-Tolerant Control with Synchronization Error

In this section, the proposed fault-tolerant control neural network-based synchronization error (PFTC NN-S-TSMC) is presented.

The ONN is adaptively used to estimate the lump of uncertainties, disturbances, and faults  $\zeta$  in Equation (13).

$$\zeta = W^T \Psi(X) + \epsilon \quad (27)$$

$X \in \mathfrak{R}^n$  is the input of the network.  $\epsilon$  is the approximation error of the neural network. The Chebyshev polynomial is used as an orthogonal function  $\Psi(X)$ . The ONN in this proposed controller has three layers:

1. The input layer:

The input vector is denoted as follows:

$$X = x_i^T = [\varepsilon_i, \dot{\varepsilon}_i]^T \quad (i = (1, 2, \dots, n)) \quad (28)$$

where  $\varepsilon_i$  and  $\dot{\varepsilon}_i$  are the synchronization error and synchronization velocity, respectively. Due to the condition of the ONN, the input must be within the range of  $[1, -1]$  so that the input can be transformed as follows:

$$x_i = \frac{2}{b-a} x_i - \frac{b-1}{b-a} \quad (29)$$

where  $a$  and  $b$  are the lower and upper bounds of  $x_i$ , respectively.

2. The hidden layer:

The Chebyshev polynomial used as an orthogonal function  $\Psi(X)$  is shown as follows:

$$\lambda_i(x) = \cos(n \arccos(x)) \quad (30)$$

It can be rewritten as follows:

$$\begin{cases} \lambda_0 = 1 \\ \lambda_1 = x \\ \lambda_2 = 2x^2 - 1 \\ \lambda_3 = 4x^3 - 3x \\ \vdots \\ \lambda_{m+1} = 2x\lambda_m(x) - \lambda_{m-1}(x) \end{cases} \quad (31)$$



3. The output layer:

The output of the ONN is expressed as follows:

$$y = W^T \Psi(X) \quad (32)$$

where  $\Psi(X)$  is shown in Equation (7).

The proposed weight update law is given as follows:

$$\dot{\hat{W}} = -\eta S^T M^{-1}(q) \Psi(X) \quad (33)$$

where  $\hat{\eta}$  is the learning rate. The proposed controller based on ONN is given as follows:

$$\tau = \tau_0 + \tau_1 + \tau_2 \quad (34)$$

where

$$\tau_0 = M(q)\dot{q}_d + H(q, \dot{q}) \quad (35)$$

$$\tau_1 = M(q)(\alpha\Lambda|\epsilon|^{\Lambda-1}\dot{\epsilon} + \beta\dot{\epsilon} + K\text{sign}(S)) \quad (36)$$

$$\tau_2 = \hat{W}^T \Psi(X) \quad (37)$$

where  $\hat{W}$  is the approximation of  $W$  in Equation (28).

**Theorem 2.** *Given the uncertain robot manipulator system in Equation (3) and the proposed controller in Equation (34) and based on the novel synchronization error in Equation (12) and sliding surface in Equation (15) with the orthogonal neural network using the weight update law in Equation (34), the tracking trajectory error of the robot manipulator converges to zero.*

**Proof of Theorem 2.** The Lyapunov function can be selected as follows:

$$V = \frac{1}{2} S^T S + \frac{1}{2} \eta \tilde{W}^T \tilde{W} \quad (38)$$

where  $\tilde{W}$  is the error approximation of  $W$ . It can be shown as follows:

$$\begin{aligned} \tilde{\zeta} &= \zeta - \hat{\zeta} \\ &= W^T \Psi(X) + \epsilon - \hat{W}^T \Psi(X) \\ &= \tilde{W}^T \Psi(X) \end{aligned} \quad (39)$$

The time derivative of  $V$  in Equation (39) is as follows:

$$\begin{aligned} \dot{V} &= S^T \dot{S} + \eta \tilde{W}^T \dot{\tilde{W}} \\ &= S^T (\ddot{\epsilon} + \alpha\Lambda|\epsilon|^{\Lambda-1}\dot{\epsilon} + \beta\dot{\epsilon}) + \eta \tilde{W}^T \dot{\tilde{W}} \\ &= S^T (\dot{q}_d - \ddot{q} + \alpha\Lambda|\epsilon|^{\Lambda-1}\dot{\epsilon} + \beta\dot{\epsilon}) + \eta \tilde{W}^T \dot{\tilde{W}} \end{aligned} \quad (40)$$

Substituting Equation (14) into Equation (39), we have the following:

$$\dot{V} = S^T (\dot{q}_d - (M(q)^{-1}(\tau - H(q, \dot{q}) - \zeta)) + \alpha\Lambda|\epsilon|^{\Lambda-1}\dot{\epsilon} + \beta\dot{\epsilon}) + \eta \tilde{W}^T \dot{\tilde{W}} \quad (41)$$

Substituting the proposed controller in Equation (34) into Equation (41), we obtain the following:

$$\dot{V} = S^T (-M^{-1}(q)(\hat{W}^T \Psi(X) - \zeta) - K\text{sign}(S)) + \eta \tilde{W}^T \dot{\tilde{W}} \quad (42)$$

Substituting Equation (28) into Equation (42) yields the following:

$$\begin{aligned}\dot{V} &= S^T M^{-1}(q)(\tilde{W}^T \Psi(X) + \epsilon) - K \text{sign}(S) + \eta \tilde{W}^T \dot{\tilde{W}} \\ &= \tilde{W}^T (S^T M^{-1}(q) \Psi(X) + \eta \dot{\tilde{W}}) + S^T M^{-1}(q) \epsilon - S^T K \text{sign}(S)\end{aligned}\quad (43)$$

Substituting the update law in Equation (33) into Equation (43), we obtain the following:

$$\begin{aligned}\dot{V} &= S^T M^{-1}(q) \epsilon - K S^T \text{sign}(S) \\ &\leq -K \|S\| + S^T M_M \epsilon\end{aligned}\quad (44)$$

$\dot{V} \leq 0$  approximately when the design parameter  $K \geq M_M \epsilon$  with the approximation error  $\epsilon$  is sufficiently small. Therefore, Theorem 2 is proven.  $\square$

## 4. Simulation Results

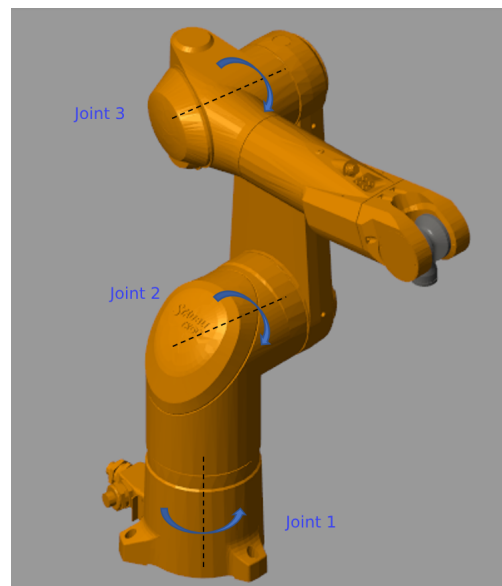
### 4.1. Simulation Setup

In this section, we present and discuss the simulation results for a conventional terminal sliding mode control and the two proposed FTC S-TSMC and FTC NN-S-TSMC controllers after applying them to a 3-DOF robot manipulator. The mechanical model of the 3-DOF robot manipulator is constructed using SolidWorks 2024 software. The geometry parameters were taken from the Staubli TX60L robot manipulator [51]. To reduce the complexity of the dynamic model calculations of the robot, joints 4, 5, and 6 of the TX60L were disabled in this study. Subsequently, the robot manipulator model was exported to a MATLAB simulation environment using the Simmechanics toolbox, as illustrated in Figure 2. The detailed parameters of the robot manipulator can be found in Table 1.

For this trajectory-tracking simulation, the desired trajectories at each joint are given as follows:

$$\begin{cases} q_{1d} = 0.5 \cos(t/2) - 0.5 \\ q_{2d} = 0.3 \cos(t/2) - 0.3 \\ q_{3d} = 0.2 \cos(t/2) - 0.2 \end{cases}\quad (45)$$

where  $q_d$  and  $\dot{q}_d$  are the first-order and second-order derivatives of the desired position, respectively.



**Figure 2.** Three-degrees-of-freedom (DOF) Staubli TX60L robot manipulator in MATLAB/Simulink with active joints.

**Table 1.** Parameters of 3-DOF Staubli TX60L robot manipulator in MATLAB simulation.

Link	Length (m)	Weight (kg)	Center of Mass (m)	Inertia (kg.m <sup>2</sup> )
Link 1	0.217	19.6	[0.05 0 0.09]	I <sub>x</sub> = [0.86 0 0.52] I <sub>y</sub> = [0 1 0] I <sub>z</sub> = [-0.52 0 0.86]
Link 2	0.4	15.2	[-0.01 0 0.1]	I <sub>x</sub> = [-0.01 0 1] I <sub>y</sub> = [0 -1 0] I <sub>z</sub> = [1 0 1]
Link 3	0.45	23.1	[0.05 0 0.08]	I <sub>x</sub> = [1 0 0.08] I <sub>y</sub> = [-0.01 1 0.08] I <sub>z</sub> = [-0.08 -0.08 0.99]

The uncertainties at each joint are assumed to be as follows:

$$\begin{cases} \Xi_1 = 0.2\text{sign}(q_1) + 0.3q_1 \\ \Xi_2 = 0.3\text{sign}(q_2) + 0.3q_2 \\ \Xi_3 = 0.32\text{sign}(q_3) + 0.2q_3 \end{cases} \quad (46)$$

The parameters in the two proposed controllers in Equations (16) and (34) are selected as  $\psi_1 = \psi_2 = \psi_3 = 2$ ,  $\alpha = \text{diag}(0.5, 0.5, 0.5)$ ,  $\beta = \text{diag}(0.72, 0.72, 0.72)$ ,  $\Lambda = \text{diag}(0.6, 0.6, 0.6)$ ,  $K = \text{diag}(80, 80, 110)$ ,  $\gamma = 0.2$ ,  $a_0 = 5$ ,  $a_1 = 20$ ,  $a_2 = 5$ ,  $\eta = 0.01$ .

The conventional fast terminal sliding mode control (FTSMC) [46] is given as follows:

$$\tau = \tau_0 + \tau_1 \quad (47)$$

where  $\tau_0 = M(q)(\ddot{q}_d + d\dot{e} + c\Lambda e^{\Lambda-1}\dot{e}) + H(q, \dot{q})$ ,  $\tau_1 = M(q)K_1\text{sign}(S)$ .

The conventional fast terminal sliding surface [46] is selected as follows:

$$S = \dot{e} + de + ce^\Lambda \quad (48)$$

The parameters for the conventional FTSMC are suitably chosen as  $c = \text{diag}(0.5; 0.5; 0.5)$ ,  $d = \text{diag}(0.72; 0.72; 0.72)$ ,  $K_1 = \text{diag}(80; 80; 110)$ ,  $\Lambda = \text{diag}(0.6; 0.6; 0.6)$ .

The active fault-tolerant control with the synchronous terminal sliding mode in [18] (AFTC S-TSMC [18]) is described as follows. The coupling position error is given as follows:

$$E = \alpha e + \beta \int [\epsilon]^\Lambda dt \quad (49)$$

where  $E = [E_1, E_2, \dots, E_n]^T \in \mathfrak{R}^n$ ,  $\alpha = \text{diag}(\alpha_i) \in \mathfrak{R}^{n \times n}$ ,  $\beta = \text{diag}(\beta_i) \in \mathfrak{R}^{n \times n}$  are positive coupling parameters, and  $\Lambda = \text{diag}(\lambda_i)$ ,  $0 < \lambda_i < 1$ .

The synchronous terminal sliding surface is given as follows:

$$S = \dot{E} + \Gamma[E]^\Lambda \quad (50)$$

where  $S = [S_1, S_2, \dots, S_n]^T \in \mathfrak{R}^n$ ,  $\dot{E} = [\dot{E}_1, \dot{E}_2, \dots, \dot{E}_n]^T \in \mathfrak{R}^n$ ,  $\Gamma = \text{diag}(\gamma_i) \in \mathfrak{R}^{n \times n}$ ,  $\gamma_i > 0$ . AFTC S-TSMC [18] is given as follows:

$$\tau = \tau_{eq} + \tau_0 + \tau_{ob} \quad (51)$$

where  $\tau_{eq} = M(q)(\ddot{q}_d + \alpha^{-1}\beta\Lambda|\epsilon|^{\Lambda-1} \cdot \dot{e} + \alpha^{-1}\Gamma\Lambda|\dot{e}|^{\Lambda-1} \cdot \dot{E}) + H(q, \dot{q})$ ,  $\tau_0 = M(q)K_1\text{sign}(S)$ ,  $\tau_{ob} = -M(q)\hat{\phi}$  which  $K_1 = \text{diag}(k_{1i}) \in \mathfrak{R}^{n \times n}$ . We obtain  $\hat{\phi}$  as follows:

$$\begin{cases} \hat{x}_1 = \hat{x}_2 + \frac{\alpha_1}{\epsilon}(x_1 - \hat{x}_1) \\ \hat{x}_2 = \hat{f}(x_1, \hat{x}_2, \tau) + \frac{\alpha_2}{\epsilon^2}(x_1 - \hat{x}_1) + \hat{\phi} \\ \hat{\phi} = \frac{\alpha_3}{\epsilon^3}(x_1 - \hat{x}_1) \end{cases} \quad (52)$$

where  $\hat{x}_1, \hat{x}_2, \hat{f}$ , and  $\hat{\phi}$  are the estimates of  $x_1 = q \in \mathbb{R}^n, x_2 = \dot{q} \in \mathbb{R}^n, f = M^{-1}(q)(\tau - H(q, \dot{q}))$ , and  $\phi = -M^{-1}(q)\zeta$ , respectively,  $\alpha_1, \alpha_2, \alpha_3$  are positive constants, and  $0 < \epsilon < 1$ . The parameters for the AFTC S-TSMC [18] were obtained as follows:  $\psi_1 = \psi_2 = \psi_3 = 2, \alpha = \text{diag}(1; 1; 1), \beta = \text{diag}(0.5; 0.5; 0.5), \Lambda = \text{diag}(0.6; 0.6; 0.6), \Gamma = \text{diag}(7; 7; 7)$ , and  $K_1 = \text{diag}(80; 80; 110)$ .

**Remark 1.** To avoid a singularity, the terms containing power  $\Lambda - I$  in Equations (16), (34), (47), and (55) are replaced with the saturation function.

$$\text{sat}(u_f, u_s) = \begin{cases} u_s & \text{if } u_f \geq u_s \\ u_f & \text{if } u_f < u_s \end{cases} \tag{53}$$

where  $u_s = 0.3$  is a positive constant, and  $u_f = \Lambda|x|^\Lambda - I \cdot \dot{x}$  with  $x = \epsilon$ .

**Remark 2.** To avoid chattering, the signum functions in Equations (16), (34), (47), and (55) are replaced with the saturation function:

$$\text{sat}(s) = \begin{cases} \text{sgn}(s)\delta & \text{if } |s| \geq \delta \\ s & \text{if } |s| < \delta \end{cases} \tag{54}$$

where  $\delta = 0.8$ .

#### 4.2. Simulation Results

In this subsection, the two kinds of actuator faults occurring at joint 2 are considered. The assumed fault function is shown as follows:

$$\begin{cases} \tau_{t1} = \tau_1 \\ \tau_{t2} = (1 - \rho_2(t))\tau_2 + f_2(t) & \text{if } t \geq 5 \\ \tau_{t3} = \tau_3 \end{cases} \tag{55}$$

where  $\rho_2(t) = 0.5 \sin(\pi t), f_2(t) = 80 \sin\left(\frac{\pi(t-5)}{2}\right)$ .

The results in Table 2 illustrate the effectiveness of the two proposed PFTC approaches. It is evident that the PFTC NN-S-TSMC significantly outperforms the two robust PFTCs. This superior performance can primarily be attributed to the online adaptability afforded by the neural network. Additionally, the AFTC S-TSMC [18] achieves a comparable performance level to that of the proposed PFTC NN-S-TSMC. This capability arises from its effective fault estimation processes, which enable the system to compensate for faults and maintain operational efficiency. However, despite AFTC S-TSMC’s competence, PFTC NN-S-TSMC retains distinct advantages over it, particularly in its ability to reduce the peaking phenomena due to the use of the PFTC scheme. This advantage can be seen in detail in Figure 3.

**Table 2.** Mean squared error  $\text{MSE} = \frac{1}{n} \sum_{i=1}^n (e_i)^2$  ( $n$  is the number of samples).

Joints	Conventional TSMC	AFTC S-TSMC [18]	PFTC S-TSMC	PFTC NN-S-TSMC
Joint 1	$1.7027 \times 10^{-8}$	$6.0705 \times 10^{-11}$	$8.4325 \times 10^{-9}$	$8.4292 \times 10^{-10}$
Joint 2	$1.5918 \times 10^{-6}$	$3.9397 \times 10^{-9}$	$1.7215 \times 10^{-8}$	$2.2357 \times 10^{-9}$
Joint 3	$2.1176 \times 10^{-6}$	$5.9448 \times 10^{-9}$	$1.0103 \times 10^{-8}$	$3.4783 \times 10^{-9}$

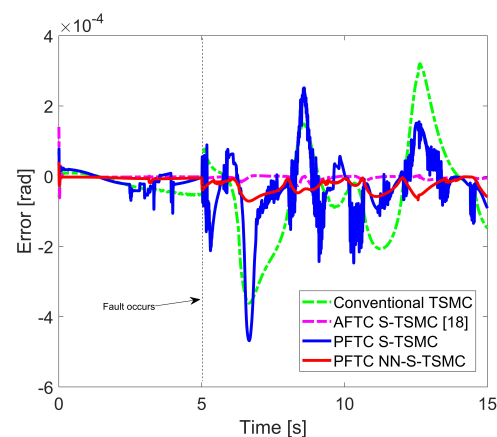
The tracking trajectory results are illustrated in Figure 3. In Figure 3a, the error levels of the four controllers are generally similar, remaining within  $10^{-4}$  radians. This joint is less affected by faults due to gravity. Notably, AFTC S-TSMC [18] exhibits superior performance compared to the other controllers due to the robustness of its fault estimation. In Figure 3b,c, prior to the fault occurrence, the position errors of all four controllers are comparable.

However, after the fault occurs at the 5th second, joints 2 and 3 are significantly impacted due to both the fault and the effects of gravity on joint 3. The conventional TSMC maintains acceptable performance, keeping the error within  $10^{-3}$  radians. Nevertheless, compared to the three proposed controllers, it still exhibits higher error levels. This difference arises because the internal constraints of the synchronization technique facilitate a rapid response, thereby mitigating the effects of the fault. When focusing on the 5th second, it is evident that the peaking phenomenon is effectively reduced in the two proposed controllers utilizing the PFTC scheme, in contrast to AFTC S-TSMC, which displays a slower response to faults due to delayed feedback from the fault estimation. Beyond this point, AFTC S-TSMC and PFTC NN-S-TSMC demonstrate similar performance due to the effectiveness of fault estimation compensation and the online adaptability of the neural network.

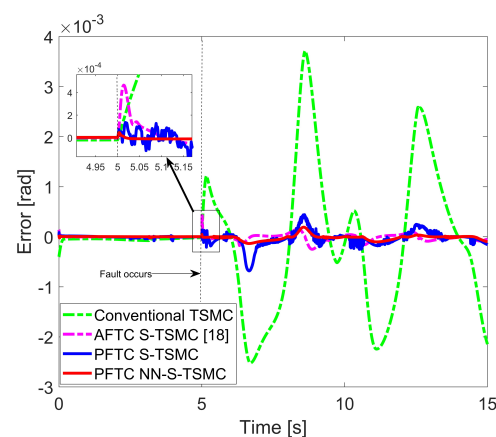
Overall, the four controllers exhibit fault tolerance. However, PFTC NN-S-TSMC outperforms the other comparative controllers.

In Figure 4, the synchronization performance is shown. In Figure 4a, the conventional TSMC does not exhibit synchronization. In Figure 4b,c, synchronization occurs. This shows that the errors at each joint tend to become similar. However, in PFTC NN-S-TSMC, the error trends at each joint occasionally differ due to the online adaptive nature of the neural network. Nevertheless, the effectiveness of the synchronization technique in the two proposed controllers is verified.

In Figure 5, the weight values of the neural network are presented. It can be seen that before the fault occurs at the 5th second, the weights nearly converged to a constant, as the synchronization errors were relatively small. However, when the fault occurred, the weights were automatically adjusted to suppress the increasing error in the system.



(a)



(b)

Figure 3. Cont.

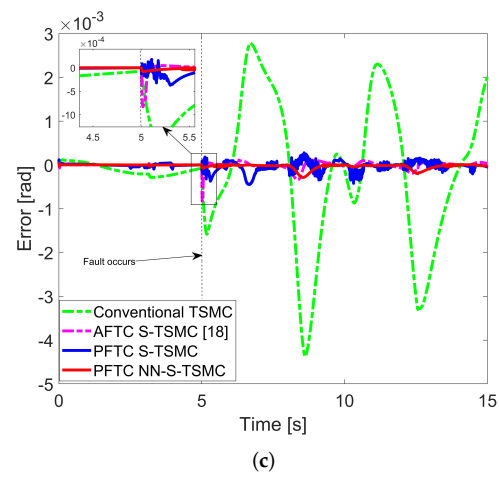


Figure 3. Tracking trajectory errors. (a) Joint 1, (b) joint 2, (c) joint 3.

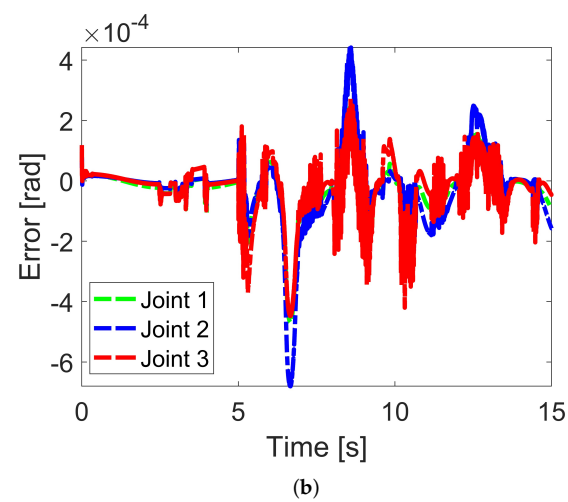
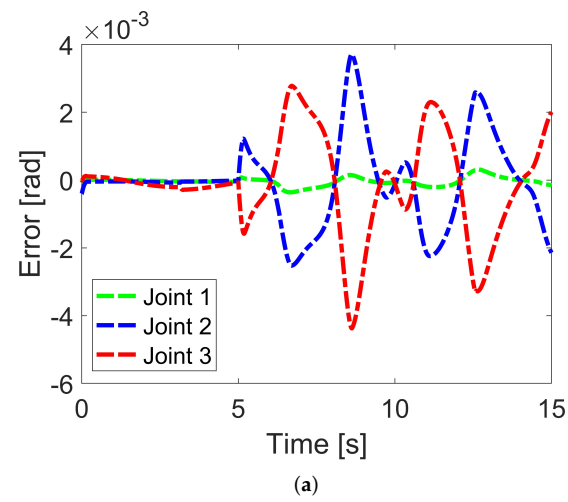
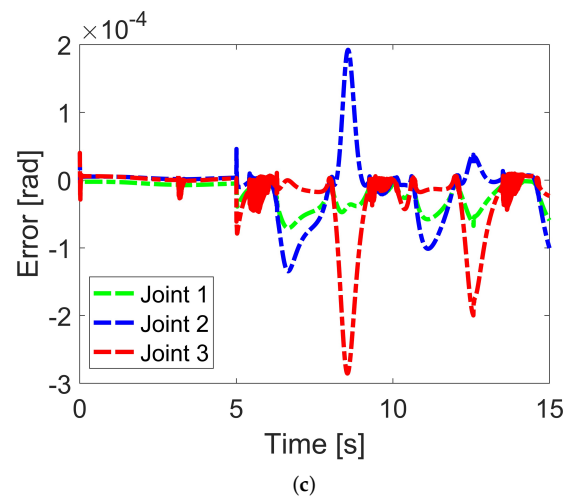
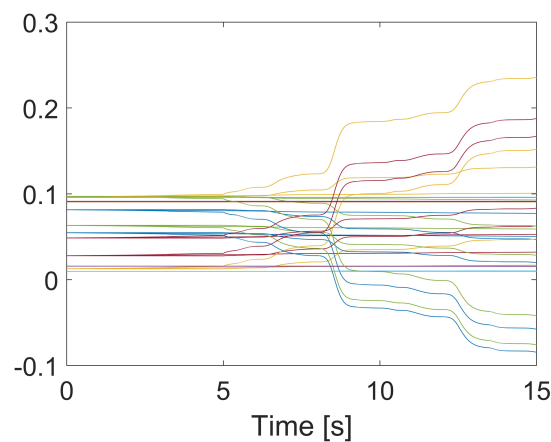


Figure 4. Cont.



**Figure 4.** Synchronization performance. (a) Conventional TSMC, (b) PFTC S-TSMC, (c) PFTC NN-S-TSMC.



**Figure 5.** The online update weights of the neural network.

## 5. Discussion and Future Work

Despite the superior fault-tolerant performance of the two proposed controllers, there are certain limitations in their designs. First, the chattering issue in sliding mode control was mitigated using a saturation function, which unfortunately increased the system's convergence time and introduced a small steady-state error. Second, selecting the control gains and coefficients for synchronization error remains a significant challenge. In this paper, the trial-and-error method used for optimizing the controller parameters is time-consuming and requires both design expertise and in-depth knowledge of the robotic system. These limitations will be addressed in future work.

To reduce chattering, higher-order sliding mode techniques, which are effective at minimizing chattering, accelerating convergence, and eliminating steady-state error, could be investigated. For the parameter selection, methods such as adaptive tuning may be employed to reduce manual effort, while intelligent techniques like neural networks or fuzzy logic could be used to automatically adjust control parameters. Furthermore, it is crucial to explore how synchronization error design parameters impact fault-tolerant control performance.

Applying the synchronization control techniques to real-world robotic systems presents several challenges, such as sensor and actuator noise, computational time, and the practical limitations of sliding mode control. These challenges must be overcome to verify the practical viability of the proposed technique. Finally, expanding the application of this control

strategy to multi-robot manipulator systems or multi-quadrotor systems in coordinated tasks will further demonstrate its broader applicability.

## 6. Conclusions

In this paper, two passive fault-tolerant controllers for a robot manipulator are proposed. The first controller, utilizing the robust synchronous terminal sliding mode control, demonstrates an advantage in reducing the peaking phenomenon when compared to active fault-tolerant control architectures. It also performs better than the conventional terminal sliding mode control. This improvement is due to the internal constraints on the joint errors imposed by the synchronization technique, which ensures that the position errors at each joint converge to zero simultaneously. This feature allows the robotic system to quickly and effectively mitigate the impact of faults, maintaining performance even when faults occur. The second proposed controller, which combines the synchronization technique with online adjustments using an orthogonal neural network, further improves fault tolerance by reducing the effects of faults more effectively than both the first proposed controller and the conventional terminal sliding mode control.

**Author Contributions:** Conceptualization, methodology, software, investigation, and writing—original draft preparation, Q.D.L.; review and editing, supervision, and project administration, E.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded in part by the SeaSense for Sustainability research project from the UK Net Zero Technology Centre and the University of Strathclyde (Grant No. 1716541, 1 October 2022–31 January 2025).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Acknowledgments:** The authors would like to thank the robotics team members at the University of Strathclyde for their kind support, especially Mohamed Khedr, Mark Robertson, and Xiutian Yan.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of this study; in the writing of manuscript; or in the decision to publish the results.

## References

1. Visinsky, M.L.; Cavallaro, J.R.; Walker, I.D. Robotic fault detection and fault tolerance: A survey. *Reliab. Eng. Syst. Saf.* **1994**, *46*, 139–158. [\[CrossRef\]](#)
2. Aldridge, H.A.; Juang, J.N. *Experimental Robot Position Sensor Fault Tolerance Using Accelerometers and Joint Torque Sensors*; Technical Report; NASA: Washington, DC, USA, 1997.
3. Le, Q.D.; Kang, H.J. Implementation of fault-tolerant control for a robot manipulator based on synchronous sliding mode control. *Appl. Sci.* **2020**, *10*, 2534. [\[CrossRef\]](#)
4. Sadeghzadeh, I.; Mehta, A.; Chamseddine, A.; Zhang, Y. Active fault tolerant control of a quadrotor uav based on gainscheduled pid control. In Proceedings of the 2012 25th IEEE Canadian conference on electrical and computer engineering (CCECE), Montreal, QC, Canada, 29 April–2 May 2012; pp. 1–4.
5. Shen, Q.; Yue, C.; Goh, C.H.; Wang, D. Active fault-tolerant control system design for spacecraft attitude maneuvers with actuator saturation and faults. *IEEE Trans. Ind. Electron.* **2018**, *66*, 3763–3772. [\[CrossRef\]](#)
6. Wang, R.; Wang, J. Passive actuator fault-tolerant control for a class of overactuated nonlinear systems and applications to electric vehicles. *IEEE Trans. Veh. Technol.* **2012**, *62*, 972–985. [\[CrossRef\]](#)
7. Benosman, M.; Lum, K.Y. Passive actuators' fault-tolerant control for affine nonlinear systems. *IEEE Trans. Control. Syst. Technol.* **2009**, *18*, 152–163. [\[CrossRef\]](#)
8. Zabihi, M.; Mehrizi, R.V.; Kasaiezadeh, A.; Pirani, M.; Khajepour, A. A Hybrid Model-Data Vehicle Sensor and Actuator Fault Detection and Diagnosis System. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 8121–8133. [\[CrossRef\]](#)
9. Berghout, T.; Benbouzid, M. Fault Diagnosis in Drones via Multiverse Augmented Extreme Recurrent Expansion of Acoustic Emissions with Uncertainty Bayesian Optimisation. *Machines* **2024**, *12*, 504. [\[CrossRef\]](#)
10. Zhang, Y.; Zang, Z.; Zhang, X.; Song, L.; Yu, Z.; Wang, Y.; Gao, Y.; Wang, L. Fault Diagnosis of Industrial Robot Based on Multi-Source Data Fusion and Channel Attention Convolutional Neural Networks. *IEEE Access* **2024**, *12*, 82247–82260. [\[CrossRef\]](#)
11. Qiu, J.; Wang, T.; Sun, K.; Rudas, I.J.; Gao, H. Disturbance Observer-Based Adaptive Fuzzy Control for Strict-Feedback Nonlinear Systems With Finite-Time Prescribed Performance. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 1175–1184. [\[CrossRef\]](#)



12. Xi, R.D.; Xiao, X.; Ma, T.N.; Yang, Z.X. Adaptive Sliding Mode Disturbance Observer Based Robust Control for Robot Manipulators Towards Assembly Assistance. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6139–6146. [[CrossRef](#)]
13. Sariyildiz, E.; Sekiguchi, H.; Nozaki, T.; Ugurlu, B.; Ohnishi, K. A Stability Analysis for the Acceleration-Based Robust Position Control of Robot Manipulators via Disturbance Observer. *IEEE/ASME Trans. Mechatronics* **2018**, *23*, 2369–2378. [[CrossRef](#)]
14. Le, Q.D.; Kang, H.J. Real Implementation of an Active Fault Tolerant Control Based on Super Twisting Technique for a Robot Manipulator. In Proceedings of the Intelligent Computing Methodologies: 15th International Conference, ICIC 2019, Nanchang, China, 3–6 August 2019; Springer: Cham, Switzerland, 2019; pp. 294–305.
15. Estrada, M.A.; Fridman, L.; Moreno, J.A. Passive fault-tolerant control via sliding-mode-based Lyapunov redesign. *IEEE Trans. Autom. Control* **2024**, *69*, 6777–6788. [[CrossRef](#)]
16. Tan, J.; Fan, Y.; Yan, P.; Wang, C.; Feng, H. Sliding Mode Fault Tolerant Control for Unmanned Aerial Vehicle with Sensor and Actuator Faults. *Sensors* **2019**, *19*, 643. [[CrossRef](#)]
17. Alwi, H.; Edwards, C. Fault Tolerant Control Using Sliding Modes with On-Line Control Allocation. In *Fault Tolerant Flight Control: A Benchmark Challenge*; Edwards, C., Lombaerts, T., Smaili, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 247–272. [[CrossRef](#)]
18. Le, Q.D.; Kang, H.J. Finite-Time Fault-Tolerant Control for a Robot Manipulator Based on Synchronous Terminal Sliding Mode Control. *Appl. Sci.* **2020**, *10*, 2998. [[CrossRef](#)]
19. Vo, A.T.; Truong, T.N.; Le, Q.D.; Kang, H.J. Fixed-Time Sliding Mode-Based Active Disturbance Rejection Tracking Control Method for Robot Manipulators. *Machines* **2023**, *11*, 140. [[CrossRef](#)]
20. Jia, F.; Huang, J.; He, X. Predefined-Time Fault-Tolerant Control for a Class of Nonlinear Systems With Actuator Faults and Unknown Mismatched Disturbances. *IEEE Trans. Autom. Sci. Eng.* **2024**, *21*, 3801–3815. [[CrossRef](#)]
21. Zhou, Y.; Liu, H.; Guo, H. L1 Adaptive Fault-Tolerant Control for Nonlinear Systems Subject to Input Constraint and Multiple Faults. *Actuators* **2024**, *13*, 258. [[CrossRef](#)]
22. Fan, H.; Fang, X.; Wang, W.; Huang, J.; Liu, L. Adaptive fault-tolerant control for uncertain nonlinear systems with both parameter estimator and controller triggering. *Automatica* **2023**, *151*, 110954. [[CrossRef](#)]
23. Le, Q.D.; Kang, H.J. An Active Fault-Tolerant Control Based on Synchronous Fast Terminal Sliding Mode for a Robot Manipulator. *Actuators* **2022**, *11*, 195. [[CrossRef](#)]
24. Koren, Y. Cross-coupled biaxial computer control for manufacturing systems. *J. Dyn. Sys. Meas. Control* **1980**, *102*, 265–272. [[CrossRef](#)]
25. Feng, L.; Koren, Y.; Borenstein, J. Cross-coupling motion controller for mobile robots. *IEEE Control Syst. Mag.* **1993**, *13*, 35–43.
26. Li, C.; Yao, B.; Wang, Q. Modeling and synchronization control of a dual drive industrial gantry stage. *IEEE/ASME Trans. Mechatronics* **2018**, *23*, 2940–2951. [[CrossRef](#)]
27. Ren, L.; Mills, J.K.; Sun, D. Experimental comparison of control approaches on trajectory tracking control of a 3-DOF parallel robot. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 982–988. [[CrossRef](#)]
28. Le, Q.D.; Kang, H.J.; Le, T.D. An Adaptive Position Synchronization Controller Using Orthogonal Neural Network for 3-DOF Planar Parallel Manipulators. In *Intelligent Computing Methodologies*; Huang, D.S., Hussain, A., Han, K., Gromiha, M.M., Eds.; Springer: Cham, Switzerland, 2017; pp. 3–14.
29. Shang, W.; Zhang, B.; Zhang, B.; Zhang, F.; Cong, S. Synchronization control in the cable space for cable-driven parallel robots. *IEEE Trans. Ind. Electron.* **2018**, *66*, 4544–4554. [[CrossRef](#)]
30. Cui, R.; Yan, W. Mutual synchronization of multiple robot manipulators with unknown dynamics. *J. Intell. Robot. Syst.* **2012**, *68*, 105–119. [[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**, *31*, 3210–3219. [[CrossRef](#)]
32. Zhang, J.; Li, S.; Xiang, Z. Adaptive fuzzy finite-time fault-tolerant control for switched nonlinear large-scale systems with actuator and sensor faults. *J. Frankl. Inst.* **2020**, *357*, 11629–11644.
33. Ma, L.; Wang, Z.; Wang, C. Adaptive neural network state constrained fault-tolerant control for a class of pure-feedback systems with actuator faults. *Neurocomputing* **2022**, *490*, 431–440. [[CrossRef](#)]
34. Zheng, X.; Shen, Q. Neural Network-Based Adaptive Fault-Tolerant Control for a Class of High-Order Strict-Feedback Nonlinear Systems. *IEEE Access* **2020**, *8*, 56510–56517. [[CrossRef](#)]
35. Jin, X.; Lü, S.; Yu, J. Adaptive NN-Based Consensus for a Class of Nonlinear Multiagent Systems With Actuator Faults and Faulty Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 3474–3486. [[CrossRef](#)]
36. Yam, J.Y.; Chow, T.W. A weight initialization method for improving training speed in feedforward neural network. *Neurocomputing* **2000**, *30*, 219–232. [[CrossRef](#)]
37. Finnoff, W.; Hergert, F.; Zimmermann, H.G. Improving model selection by nonconvergent methods. *Neural Netw.* **1993**, *6*, 771–783. [[CrossRef](#)]
38. Qian, N. On the momentum term in gradient descent learning algorithms. *Neural Netw.* **1999**, *12*, 145–151. [[CrossRef](#)] [[PubMed](#)]
39. Kayacan, E.; Khanesar, M.A. Chapter 5—Gradient Descent Methods for Type-2 Fuzzy Neural Networks. In *Fuzzy Neural Networks for Real Time Control Applications*; Kayacan, E., Khanesar, M.A., Eds.; Butterworth-Heinemann: Oxford, UK, 2016; pp. 45–70. [[CrossRef](#)]

40. Yang, S.S.; Tseng, C.S. An orthogonal neural network for function approximation. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 779–785. [[CrossRef](#)]
41. Li, S.; Jia, K.; Wen, Y.; Liu, T.; Tao, D. Orthogonal Deep Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1352–1368. [[CrossRef](#)]
42. Wang, J.; Chen, Y.; Chakraborty, R.; Yu, S.X. Orthogonal Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
43. Mashhadi, P.S.; Nowaczyk, S.; Pashami, S. Parallel orthogonal deep neural network. *Neural Netw.* **2021**, *140*, 167–183. [[CrossRef](#)]
44. Perić, S.L.; Antić, D.S.; Milovanović, M.B.; Mitić, D.B.; Milojković, M.T.; Nikolić, S.S. Quasi-Sliding Mode Control With Orthogonal Endocrine Neural Network-Based Estimator Applied in Anti-Lock Braking System. *IEEE/ASME Trans. Mechatronics* **2016**, *21*, 754–764. [[CrossRef](#)]
45. Le, Q.D.; Kang, H.J.; Le, T.D. An Adaptive Controller with An Orthogonal Neural Network and A Third Order Sliding Mode Observer for Robot Manipulators. *Int. J. Mech. Eng. Robot. Res.* **2016**, *7*, 169–173. [[CrossRef](#)]
46. Yu, X.; Zhihong, M. Fast terminal sliding-mode control design for nonlinear dynamical systems. *IEEE Trans. Circuits Syst. Fundam. Theory Appl.* **2002**, *49*, 261–264.
47. Liu, H.; Zhang, T. Neural network-based robust finite-time control for robotic manipulators considering actuator dynamics. *Robot. Comput.-Integr. Manuf.* **2013**, *29*, 301–308. [[CrossRef](#)]
48. Rodd, M.G. *Introduction to Robotics: Mechanics and Control*; Craig, J.J., Ed.; Prentice Hall: Englewood Cliffs, NJ, USA, 1987.
49. Bošković, J.D.; Mehra, R.K. Failure detection, identification and reconfiguration system for a redundant actuator assembly. *IFAC Proc. Vol.* **2003**, *36*, 411–416. [[CrossRef](#)]
50. Courant, R.; Hilbert, D.R. Methods of Mathematical Physics. *Math. Gaz.* **1947**, *31*, 174–176.
51. Chen, X.; Zhang, Q.; Sun, Y. Non-kinematic calibration of industrial robots using a rigid-flexible coupling error model and a full pose measurement method. *Robot. Comput.-Integr. Manuf.* **2019**, *57*, 46–58. [[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.