mathematics

MDPI

Article

Fault-Tolerant Nonlinear Control of Robotic Manipulator Using Neural Network

Humam Al-Baidhani [1,2,*], Adel A. Eidan [1] and Marian K. Kazimierczuk [2]

[1] Department of Mechanical Power Engineering Techniques, Technical Engineering College of Al-Najaf, Al-Furat Al-Awsat Technical University (ATU), Al-Najaf 54001, Iraq; inj.adel@atu.edu.iq
[2] Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA; marian.kazimierczuk@wright.edu
* Correspondence: al-baidhani.2@wright.edu

Abstract

This research aims to present a systematic approach to design a robust fault-tolerant nonlinear controller for two-link robotic manipulators to track the desired trajectory in the presence of modeling uncertainties and unknown faults. A conventional nonlinear controller is utilized as a baseline control law to design an intelligent control scheme with fault detection, dynamic estimation, and robust tracking capabilities. An adaptive scheme is developed to estimate the states under modeling uncertainties. The stability analysis of the adaptive scheme is provided to ensure the tracking error convergence and boundedness. In addition, an adaptive threshold is designed to detect the nonlinear fault that occurs in the robotic system dynamics, and an estimation model is developed using an online approximator to estimate the nonlinear fault. The proposed control system is constructed and simulated in MATLAB/SIMULINK to validate the design approach. The simulation results show that the intelligent control system provides accurate tracking performance, maintains the system stability, and accommodates the nonlinear fault under modeling uncertainty.

Keywords: fault detection; intelligent control; online approximator; robotic manipulator; dynamic estimation model; modeling uncertainty

MSC: 68T40; 70E60; 93C10

1. Introduction

The advancement in robotic technology has increased tremendously in recent decades [1]. The progress is attributed to the broad applications of robotic manipulators in various fields that require precise tracking performance and fast dynamical response. For instance, manufacturing processes, medical treatments, industrial automation, aerospace technology, and rescue missions depend on versatile robotic manipulators to accomplish critical tasks precisely and safely [2]. However, it is known that robotic manipulators are nonlinear systems, which require proper control strategies to enhance the transient response, track the desired trajectories, and maintain system stability. In addition, the modeling uncertainties, external disturbances, and nonlinear faults that the robotic manipulators encounter are unavoidable, which deteriorate the tracking performance and affect the system stability [3,4]. Hence, advanced control techniques are required for the robotic manipulators in order to achieve robust tracking performance, maintain system stability, and handle the system nonlinearities under failure conditions and unknown system dynamics.

Check for updates

Academic Editors: Marta Covadonga Mora Aguilar and Joaquin Cerda Boluda

Received: 20 August 2026
Revised: 1 October 2026
Accepted: 8 October 2026
Published: 29 October 2026

Copyright: © 2026 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.

Various control methods have been proposed in the literature to control the robotic manipulators and achieve the desired tracking performance. For instance, an error transformation embedded with a Lyapunov-Krasovskii functional (LKF) has been presented to control a two-link robotic manipulator under unknown nonlinearities and time-varying input delays [5]. In [6], an approximate constraint-following control method has been developed to obtain precise trajectory tracking with modeling uncertainties and external disturbances. A decentralized controller–observer has also been introduced in [7] for a two-degree-of-freedom (2-DoF) open-chain robotic manipulator, in which semiglobal exponential convergence of tracking error is achieved. An event-triggered control scheme combined with an adaptive super-twisting algorithm has been proposed in [8] for a 2-DoF robotic manipulator. A control design approach for a 2-DoF robotic manipulator under model uncertainties and external disturbances using a nonlinear functional observer has been introduced in [9], which can estimate lumped disturbances, uncertainties, and unmeasurable joint velocities.

Furthermore, an optimal controller [10] and a hybrid decoupling constrained controller with a proportional controller [4] have been designed for 2-DoF robots to provide robust tracking performance. Other researchers have presented robust PID-based control methods to solve tracking problems. For example, an optimal PID controller based on the solution of a mixed H2/H_∞ tracking control law [11], a PD-type iterative learning control algorithm with a forgetting factor [12], a fractional-order PID control scheme [13], a neural fractional-order PID controller [14], and a hybrid PD-PI with a fuzzy controller [3] have been presented. A digital PID control scheme has been developed and implemented for an electromechanical structure using a real-time operating system [15]. Moreover, intelligent control techniques such as neural-network-based control [16,17], finite-time robust recurrent neural-network control [18], and adaptive fuzzy logic control [19,20] have been developed.

On the other hand, nonlinear control approaches have been proposed for robotic systems. In [21], a finite-time adaptive visual tracking control for robotic manipulators with modeling uncertainties has been developed. A backstepping controller has been discussed in [22] to track the desired trajectory of a two-link robotic manipulator in the presence of noise. Other research works have applied the sliding-mode control (SMC) technique to ensure robust tracking performance. Nonlinear control methods have included SMC [23], adaptive disturbance observer-based SMC [24], fast terminal SMC [25], and adaptive SMC with fuzzy disturbance observer [26]. In addition, fast terminal SMC with time delay estimation [27], adaptive dynamic bounded SMC with disturbance observer [28], fractional SMC [29], and adaptive SMC [2] have been studied. Despite the robustness of the SMC technique, which is suitable for variable structure systems, the main issues include the chattering phenomenon, variable switching frequency, and steady-state error [30]. Therefore, an appropriate design approach should be adopted to derive the SMC law and avoid such issues. For instance, the equivalent control method is used to suppress the chattering effect and maintain a constant switching frequency [31].

The aforementioned control techniques have provided precise tracking performance in the presence of modeling uncertainties and external disturbances. However, other research works have been directed to develop fault-tolerant control (FTC) schemes to ensure accurate tracking performance and safe operation despite the faults that occur in the robot components, such as actuator failure and sensor faults [32–36]. Various FTC systems have been devised for robotic systems. For instance, FTC techniques have introduced adaptive control [37,38], fuzzy adaptive control [39], intelligent control [40], backstepping and neural-network control [41], and time-synchronized convergence control [42]. In addition, robust FTC systems have been presented using computed-torque control [43], optimal control [44], SMC [45,46], adaptive SMC [47], and neural-network-based FTC [48].

Recent research endeavors have presented adaptive FTC using fractional-order dynamic surface control with backstepping [49], sliding-mode surface combined with active optimal control [50], and integrated FTC with H_∞ optimization [51].

The previous efforts have developed robust fault-tolerant control schemes for robotic systems under unknown faults, modeling uncertainties, and disturbances. However, a systematic design procedure for an intelligent FTC system for a robotic manipulator that maintains robust tracking performance has not been introduced. Table 1 presents a brief comparison between the proposed FTC method and other recent approaches.

Table 1. Comparison of the proposed FTC method with recent research endeavors.

Criteria	You et al. [49]	Yue et al. [50]	El Fezazi et al. [51]	Proposed FTC Method
Fault detection/estimation	Passive/Active integration via online parameter boundary estimation.	Active fault estimation observer directly integrated into the control law.	Active simultaneous robust fault estimation using H_∞ optimization.	Adaptive threshold combined with dynamic estimation model.
Handling of nonlinearities and uncertainties	Fuzzy logic systems approximate unknown nonlinearities; auxiliary function bound estimators handle disturbances.	Handled via adaptive dynamic programming and NN.	T–S fuzzy model blends local linear models with H_∞ optimization.	PD computed-torque control with a sigmoid neural network.
Compensation mechanism	Adaptive FTC using fractional-order dynamic surface control (DSC) and backstepping.	Sliding-mode surface combined with active optimal compensation.	Integrated FTC acting on real-time vehicle lateral dynamics states.	Nonlinear FTC using computed-torque control, fault detection, and NN online approximator
Stability and convergence guarantee	Finite-Time Stability using a modified fractional differential inequality lemma.	Uniform ultimate boundedness under policy iteration and Lyapunov validation.	Asymptotic stability formulated via Linear Matrix Inequalities (LMI).	Asymptotic stability via Lyapunov method and Barbalat's Lemma.
Implementation complexity	Medium-High: Backstepping explosion avoided via DSC, but fractional-order calculations add difficulty.	Medium: replaces standard Actor-Critic NNs with a Critic-Only NN architecture.	Medium: heavy offline calculation (LMI solving), but highly efficient with online matrix execution.	Low-Medium: offline NN training requirement for robust tracking performance.
Detailed design procedure	Not provided	Not provided	Not provided	Provided

Hence, motivated by the design approach in [48], this research aims to design, analyze, and validate an intelligent fault-tolerant control scheme for a 2-link robotic manipulator using an online approximator. The proposed nonlinear control system contains an adaptive threshold and an identification model, which are designed to estimate the state variables and detect the nonlinear fault in the system. The stability analysis is provided to ensure the tracking error convergence and boundedness. In addition, a dynamic estimation model using a neural network is developed so that the conventional nonlinear controller can track the desired trajectory under modeling uncertainties and unknown faults.

The research contributions are summarized as follows:

1. A state-estimation model is designed to estimate the state variables of the robotic system under modeling uncertainties.

2. An adaptive threshold is designed to detect the nonlinear fault in a 2-link robotic manipulator under modeling uncertainties.
3. A dynamic identification model is developed using a neural network to approximate the nonlinear fault online.
4. Stability analysis of the closed-loop FTC system is provided, which incorporates the tracking error and NN weight estimation error.
5. A robust fault-tolerant control scheme is constructed using MATLAB/SIMULINK, so that the nominal nonlinear controller can compensate for unknown faults and track the desired trajectory accurately.

The paper is organized as follows. The nonlinear model and the nominal controller of the 2-link robotic manipulator are given in Section 2. The design procedure of the intelligent control system, which includes the identification model, adaptive threshold, and online approximator, is presented in Section 3. The simulation results and discussion are introduced in Section 4. The conclusions are summarized in Section 5.

2. Nonlinear Model of Robotic Manipulator

2.1. Dynamics of Robotic Manipulator

According to [52], the dynamics of robot manipulators with rigid links can be written as

$$M(q)\ddot{q} + V_m(q,\ \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = \tau \tag{1}$$

where $M(q)$ is the inertia matrix, $V_m(q,\dot{q})$ is the Coriolis/centripetal matrix, $F(\dot{q})$ are the friction terms, $G(q)$ is the gravity vector, $\tau_d(t)$ represents disturbances, and $\tau(t)$ is the control input vector.

The robot dynamics can also be written in a more convenient form as

$$M(q)\ddot{q} + N(q,\ \dot{q}) + \tau_d = \tau \tag{2}$$

where the vector of nonlinear terms $N(q,\dot{q})$ are given by

$$N(q,\ \dot{q}) = V_m(q,\ \dot{q})\dot{q} + F(\dot{q}) + G(q) \tag{3}$$

The 2-link robotic manipulator is depicted in Figure 1. According to Lagrange's equation, the robotic arm dynamics are given by two coupled nonlinear differential equations

$$\begin{aligned}\tau_1 = [(m_1+m_2)a_1^2 \quad & +m_2a_2^2 + 2m_2a_1a_2cos\ q_2]\ddot{q}_1 + \left[m_2a_2^2 + m_2a_1a_2cosq_2\right]\ddot{q}_2 \\ & -m_2a_1a_2\left(2\dot{q}_1\dot{q}_2 + \ddot{q}_2^2\right)sin\ q_2 + (m_1+m_2)ga_1\cos q_1 \\ & +m_2ga_2cos(q_1+q_2)\end{aligned} \tag{4}$$

$$\tau_2 = \left[m_2a_2^2 + m_2a_1a_2cos\ q_2\right]\ddot{q}_1 + m_2a_2^2\ddot{q}_2 + m_2a_1a_2\dot{q}_1^2sin\ q_2 + m_2ga_2cos(q_1+q_2) \tag{5}$$

where m_1 and m_2 represent the mass of link 1 and 2, respectively. The parameters a_1 and a_2 represent the length of link 1 and 2, respectively. In addition, q represents the joint angle vector, g is the gravity of Earth, and τ is the generalized force vector.

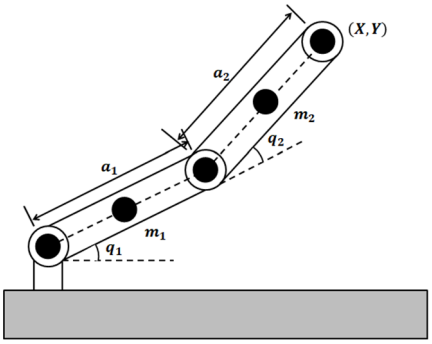


Figure 1. 2-link robotic manipulator.

2.2. State-Space Model of Robotic Manipulator

The nonlinear state-space representation is convenient for the control law derivation and MATLAB R2022b simulation. Since there are two 2nd order differential equations for the system dynamics, four state variables are required to describe the state-space model. In [52], the nonlinear state-space model of the 2-link robotic arm is

$$\begin{cases} \dot{q}_1 = q_3 \\ \dot{q}_2 = q_4 \\ \dot{q}_3 = M_{11}^{-1}(-N_1 + u_1) + M_{12}^{-1}(-N_2 + u_2) \\ \dot{q}_4 = M_{12}^{-1}(-N_1 + u_1) + M_{22}^{-1}(-N_2 + u_2) \end{cases} \tag{6}$$

where the state variables q_1 and q_2 are the joint angles of link 1 and 2, respectively. The time derivatives $\dot{q}_1$ and $\dot{q}_2$ are the velocities of link 1 and 2, respectively.

The elements M_{11}^{-1}, M_{12}^{-1}, and M_{22}^{-1} are defined as follows [52]

$$\begin{cases} M_{11}^{-1} = \frac{M_{22}}{M_{11}M_{22} - M_{12}^2} \\ M_{12}^{-1} = \frac{-M_{12}}{M_{11}M_{22} - M_{12}^2} \\ M_{22}^{-1} = \frac{M_{11}}{M_{11}M_{22} - M_{12}^2} \end{cases} \tag{7}$$

It should also be noted that M_{11}, M_{12}, M_{22}, N_1 and N_2 are obtained from the nonlinear coupled differential equations, which are given by

$$\begin{cases} M_{11} = (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2cos(\ q_2) \\ M_{12} = m_2a_2^2 + m_2a_1a_2cos(q_2) \\ M_{22} = m_2a_2^2 \\ N_1 = -m_2a_1a_2\left(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2\right)sin(q_2) + (m_1 + m_2)ga_1\cos(q_1) + m_2ga_2cos(q_1 + q_2) \\ N_2 = m_2a_1a_2\dot{q}_1^2sin(q_2) + m_2ga_2cos(q_1 + q_2) \end{cases} \tag{8}$$

The parameters of the manipulator are a_1, a_2, m_1, and m_2 are 1 m, 1 m, 0.5 kg, and 2 kg, respectively [48].

In this research, the modeling uncertainties and disturbance vectors are defined as follows

$$F(\dot{q}) = \begin{bmatrix} 0.45\dot{q}_1 \\ 0.45\dot{q}_2 \end{bmatrix} \tag{9}$$

$$\tau_d = \begin{bmatrix} 0.4sin(30t) \\ 0.4sin(30t) \end{bmatrix} \tag{10}$$

while the desired trajectory vector of the robotic system is given by

$$q_d = \begin{bmatrix} 0.2\sin\left(\frac{2\pi}{6}t\right) \\ 0.2\cos\left(\frac{2\pi}{6}t\right) \end{bmatrix} \tag{11}$$

2.3. Nominal Control Law Performance

The robotic system model has a nonlinear part, which requires a nonlinear control technique to obtain an output that tracks the desired trajectory. The nominal control law has been adopted from [48]. Using the computed-torque method, the control law is written as

$$\tau = M(q)\left(\ddot{q}_d + K_d\dot{e} + K_pe\right) + V_m(q,\ \dot{q})\dot{q} + G(q) \tag{12}$$

The gains K_p and K_d are the derivative and proportional gains, respectively. In [52], the gains are selected as $K_p = 100$, and $K_d = 20$. The tracking error is defined as $e = q_d - q$.

The gains of the control law have to be set such that the tracking error converges exponentially to zero. However, due to the modeling uncertainties and disturbance terms,

a slight deviation from the desired trajectory and a bounded tracking error appear in the robotic system response. Figure 2 shows the MATLAB simulation of the joint angle tracking error of the computed-torque-controlled robotic system under modeling uncertainties and disturbances.

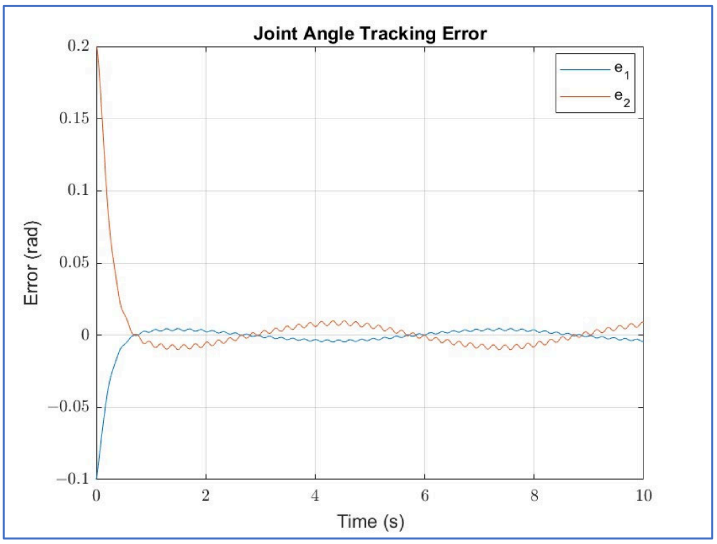


Figure 2. Joint angle tracking error of the computed-torque-controlled robotic system under modeling uncertainties.

3. Design of Fault-Tolerant Control Scheme

3.1. Robotic System Dynamics Under Fault Effect

Based on the previous section, if bounded modeling uncertainties and disturbances are included, the system dynamic equations yield

$$\begin{cases} \dot{q}_1 = q_3 \\ \dot{q}_2 = q_4 \\ \dot{q}_3 = M_{11}^{-1}\left(-N_1 + u_1 - \theta_1\dot{q}_1 - \theta_2 sin(30t)\right) \\ \quad\quad +M_{12}^{-1}\left(-N_2 + u_2 - \theta_1\dot{q}_2 - \theta_2 sin(30t)\right) \\ \dot{q}_4 = M_{12}^{-1}\left(-N_1 + u_1 - \theta_1\dot{q}_1 - \theta_2 sin(30t)\right) \\ \quad\quad +M_{22}^{-1}\left(-N_2 + u_2 - \theta_1\dot{q}_2 - \theta_2 sin(30t)\right) \end{cases} \tag{13}$$

where the unknown parameters are $\theta_1 \in [0\ 0.5]$ and $\theta_2 \in [0\ 0.5]$.

Now, if a fault is applied to the dynamics given in (13), the robotic system model is written in the following form

$$\begin{cases} \dot{q}_1 = q_3 \\ \dot{q}_2 = q_4 \\ \dot{q}_3 = M_{11}^{-1}\left(-N_1 + u_1 - \theta_1\dot{q}_1 - \theta_2 sin(30t)\right) \\ +M_{12}^{-1}\left(-N_2 + u_2 - \theta_1\dot{q}_2 - \theta_2 sin(30t)\right) + F(1) \\ \dot{q}_4 = M_{12}^{-1}\left(-N_1 + u_1 - \theta_1\dot{q}_1 - \theta_2 sin(30t)\right) \\ +M_{22}^{-1}\left(-N_2 + u_2 - \theta_1\dot{q}_2 - \theta_2 sin(30t)\right) + F(2) \end{cases} \tag{14}$$

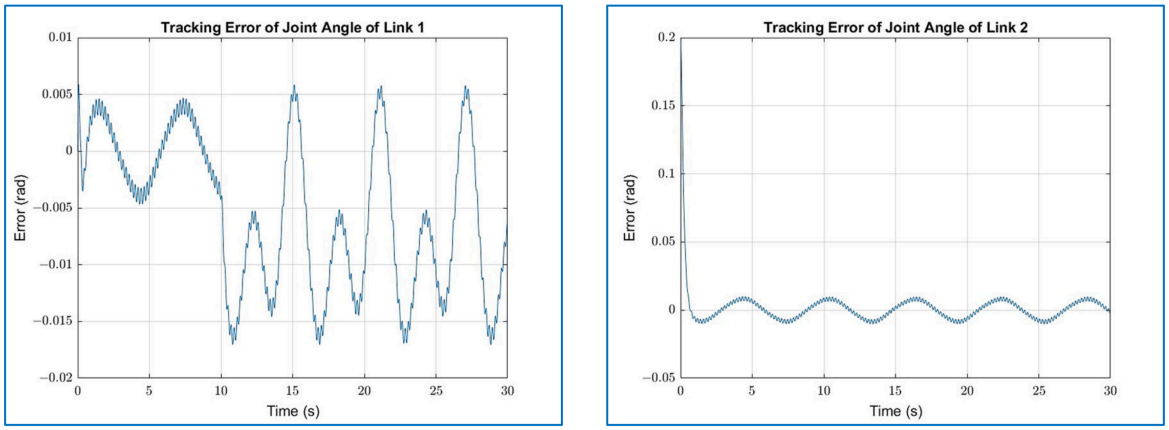
where the fault vector is assumed to be

$$F = \begin{bmatrix} 37.5(q_1)^2 + 50(q_1)^2(q_2)^2 + 3.5q_2 \\ 0 \end{bmatrix} \tag{15}$$

The fault manifests as a combination of structural deformation, joint binding, and nonlinear mechanical cross-coupling. This fault causes steady-state position offset and

trajectory divergence, as the nominal controller lacks the model terms to compensate for this highly nonlinear fault.

MATLAB simulation is conducted to investigate the tracking performance of the nominal nonlinear controller under modeling uncertainty and nonlinear fault. In Figure 3, it can be observed that the tracking error of the joint angle of link 1 increases significantly when the fault vector in (15) is applied to the robotic system at $t \geq 10$ s, while the tracking error of the joint angle of link 2 is unaffected.



(**a**) (**b**)

Figure 3. Joint angle tracking error of (**a**) link 1 and (**b**) link 2 in the computed-torque controlled robotic system under modeling uncertainties and nonlinear fault effect.

Hence, an intelligent control scheme is required to detect and approximate the nonlinear fault, so that the nominal nonlinear controller can tolerate the fault and maintain an accurate tracking performance.

3.2. Design of Identification Model

If a nonlinear fault is subjected to the system dynamics, then a suitable identification model with an adaptive threshold is required to detect the fault. In this research, the series-parallel scheme is selected to derive the adaptive law for the robotic system with modeling uncertainty.

Let us assume that the robotic system model $\dot{q}$ is defined by the following expression

$$\dot{q} = \varphi(q) + \eta(q, u, t) \tag{16}$$

where $\varphi(q)$ is the nominal model of the system and $\eta(q,u,t)$ is the modeling uncertainty. The estimation of the system dynamics $\dot{\hat{q}}$ can be written as

$$\dot{\hat{q}} = -a_m(\hat{q} - q) + \varphi(q) \tag{17}$$

where a_m is a positive number greater than zero, and $\hat{q}$ is the state estimation. Substituting the nominal model of the manipulator in (17) yields

$$\begin{cases} \dot{\hat{q}}_1 = -a_m(\hat{q}_1 - q_1) + q_3 \\ \dot{\hat{q}}_2 = -a_m(\hat{q}_2 - q_2) + q_4 \\ \dot{\hat{q}}_3 = -a_m(\hat{q}_3 - q_3) + M_{11}^{-1}(-N_1 + u_1) + M_{12}^{-1}(-N_2 + u_2) \\ \dot{\hat{q}}_4 = -a_m(\hat{q}_4 - q_4) + M_{12}^{-1}(-N_1 + u_1) + M_{22}^{-1}(-N_2 + u_2) \end{cases} \tag{18}$$

The adaptive laws in (18) are used to estimate the state variables of the system dynamics under modeling uncertainty. Then, the state-estimation vector will be used to obtain the state-estimation error, which should have a bounded magnitude due to the modeling uncertainty.

3.3. Design of Adaptive Threshold

The nonlinear system dynamics of the robotic manipulator under fault can be written in compact form as

$$\dot{q} = \varphi(q,\ u) + \eta(q, u,\ t) + \beta(t - T_0) f(q, u) \tag{19}$$

where q is a measurable state vector, u is a control input, $\varphi(q,\ u)$ is the known nominal system dynamics, $\eta(q,u,\ t)$ is modeling uncertainty. $\beta(t - T_0)$ is a fault time profile with an unknown fault occurrence time T_0, and $f(q,u)$ is an unknown fault function.

During an abrupt fault, $\beta(t - T_0)$ takes the form of a step function

$$\beta(t - T_0) = \begin{cases} 0 \text{ for } t < T_0 \\ 1 \text{ for } t \ \geq \ T_0 \end{cases} \tag{20}$$

Assuming that the modeling uncertainty is bounded by a known function

$$|\eta(q, u,\ t)| \leq \overline{\eta}(q, u,\ t) \tag{21}$$

Before fault occurrence, $\beta(t - T_0)$ is zero; therefore,

$$\dot{q} = \varphi(q,\ u) + \eta(q, u,\ t) \tag{22}$$

Let the state-estimation error be expressed as $\tilde{q} = q - \hat{q}$. Then, the state-estimation error dynamics yield $\dot{\tilde{q}} = \dot{q} - \dot{\hat{q}}$, which can also be written as

$$\dot{\tilde{q}} = -a_m \tilde{q} + \eta(q, u,\ t) \tag{23}$$

The solution of the differential equation given in (23) is

$$\tilde{q}(t) = e^{-a_m t}\ \tilde{q}(0) + \int_0^t e^{-a_m(t-\tau)}\ \eta(q, u,\ \tau)\ d\tau \tag{24}$$

Furthermore,

$$|\tilde{q}(t)| = \left| e^{-a_m t}\ \tilde{q}(0) + \int_0^t e^{-a_m(t-\tau)}\ \eta(q, u,\ \tau)\ d\tau \right|$$

Using the triangular inequality property, the following is obtained

$$|\tilde{q}(t)| \leq \varepsilon_0 e^{-a_m t} + \int_0^t e^{-a_m(t-\tau)}\ |\eta(q, u,\ \tau)\ d\tau| \tag{25}$$

Also,

$$|\tilde{q}(t)| \leq \varepsilon_0 e^{-a_m t} + \int_0^t e^{-a_m(t-\tau)}\ \overline{\eta}(q, u,\ \tau)\ d\tau$$

where ε_0 is the initial error, $\overline{\eta}(q, u,\ \tau)$ is the upper bound of the modeling uncertainty, and a_m is a constant greater than zero. If the initial estimation error is assumed to be zero, the term $\varepsilon_0 e^{-a_m t}$ can be eliminated. Hence, the fault detection threshold $V(t)$ can be chosen as

$$V(t) = \int_0^t e^{-a_m(t-\tau)}\ \overline{\eta}(q, u,\ \tau) d\tau \tag{26}$$

It can be seen that when $\beta(t - T_0)$ is zero, it is guaranteed that $|\tilde{q}(t)| \leq V(t)$.

For the robotic manipulator with the given modeling uncertainty, the threshold is implemented as follows

$$\begin{cases} \dot{\tilde{q}}_1 = -a_m\ \tilde{q}_1 \\ \dot{\tilde{q}}_2 = -a_m\ \tilde{q}_2 \\ \dot{\tilde{q}}_3 = -a_m\ \tilde{q}_3 + \theta_1 \dot{q}_1 + \theta_2 \\ \dot{\tilde{q}}_4 = -a_m\ \tilde{q}_4 + \theta_1 \dot{q}_2 + \theta_2 \end{cases} \tag{27}$$

Based on the assumption in (21), the upper bound of the parameters θ_1 and θ_2 can be used in the threshold, where θ_1 and $\theta_2 \in [0, 0.5]$. The expression in (27) guarantees that the residual does not exceed the threshold during normal operating condition. The adaptive threshold should have two important properties:

1. No false alarms are produced. Although the residual signal takes a small value around zero, it should not be detected as a fault.
2. No missed detection occurs. The adaptive threshold should detect the fault once the residual signal exceeds the bounded modeling uncertainty.

3.4. Design of Dynamic Estimation Model

The next step is to design a dynamic estimation model using online approximators, such as a sigmoid neural network, so that the unknown fault can be accommodated by the feedback nominal controller.

Let us assume that the dynamics of the robotic manipulator are given by

$$\dot{q} = \zeta(q,\ u) + f(q,\ u) \tag{28}$$

where $\zeta(q,\ u)$ is the known part of the robotic model and $f(q,\ u)$ is the unknown fault function. Then, the dynamic estimation model can be written as

$$\dot{\hat{q}} = -a_m \hat{q} + \zeta(q,\ u) + a_m q + \hat{f}\left(q,\ u,\ \hat{\theta}\right) \tag{29}$$

where $\hat{f}\left(q,\ u,\ \hat{\theta}\right)$ is an online approximator, $\hat{\theta}$ is an adjustable weight, and $a_m > 0$ is a design constant.

The adaptive law that is used to update the weights is given by

$$\dot{\hat{\theta}} = \Gamma\, Z^T \varepsilon \tag{30}$$

where the gradient is $Z = \frac{\partial \hat{f}}{\partial \hat{\theta}}\left(q,\ u,\ \hat{\theta}\right)$, the state-estimation error $\varepsilon = q - \hat{q}$, and the learning rate $\Gamma > 0$ is a positive definite diagonal matrix.

For the SNN approximator, 12 adjustable weights θ have been set for each state. Since the robotic manipulator has 4 states, 48 weights are required. However, since the fault is injected to $\dot{q}_1$ as described in (14), only 12 weights are involved in the learning and approximation process. The SNN model with 4 nodes and 12 adjustable weights can be expressed as

$$\hat{f}(q,\ \theta_1, \theta_2,\ \ldots,\ \theta_{12}) = \sum_{i=1}^{4} \hat{\theta}_i\, \sigma\left(\hat{\theta}_{i+4}\, q + \hat{\theta}_{i+8}\right) \tag{31}$$

In addition, the dynamic estimation model using a series-parallel scheme is given by

$$\dot{\hat{q}} = -a_m \hat{q} + a_m q + \sum_{i=1}^{4} \hat{\theta}_i\, \sigma\left(\hat{\theta}_{i+4}\, q + \hat{\theta}_{i+8}\right) + u \tag{32}$$

whereas the adaptive laws that are used to update the weights for one state are

$$\dot{\hat{\theta}}_i = \gamma_i\, \sigma(p_i)\ (q - \hat{q}),\ i = 1,\ \ldots\ ,\ 4 \tag{33}$$

$$\dot{\hat{\theta}}_{i+4} = \gamma_i\, \hat{\theta}_i\, q\, e^{-p_i}\, \sigma^2(p_i)\ (q - \hat{q}),\ i = 5,\ \ldots\ ,\ 8 \tag{34}$$

$$\dot{\hat{\theta}}_{i+8} = \gamma_i\, \hat{\theta}_i\, e^{-p_i}\, \sigma^2(p_i)\ (q - \hat{q}),\ i = 9,\ \ldots\ ,\ 12 \tag{35}$$

The terms $\sigma(p_i)$ and (p_i) are defined as follows

$$\sigma(p_i) = \frac{1}{1 + e^{-p_i}} \tag{36}$$

$$p_i = \hat{\theta}_{i+4}\, q + \hat{\theta}_{i+8} \tag{37}$$

where γ_i is the learning rate, q and $\hat{q}$ are the state and the state estimation of the dynamic system, respectively. In the simulation, the values of the parameters γ and a_m are set to 25 and 2, respectively.

The SNN estimation model can be modeled using a MATLAB function, where 52 states are presented. In other words, there are 48 adjustable weights for the network (12 for each state) and 4 states for the system dynamics.

Among all the 52 outputs of the SNN estimation model, only 12 weights are relevant to $\dot{q}_1$, which are used for the approximated function $\hat{f}(x)$ (i.e., the weights θ_{25}, ..., θ_{36}). The MATLAB function of $\hat{f}(x)$ is used to approximate the unknown fault function after the fault detection. The learning capability of the neural network should be quite enough to accommodate the fault within a short time frame.

3.5. Stability Analysis of Closed-Loop Control System

Let the nonlinear system dynamics of the robotic manipulator with unknown nonlinear fault $f(q, u)$ be written as

$$\dot{q} = \zeta(q,\ u) + f(q,\ u) \tag{38}$$

and the robotic system is augmented with a dynamic estimation model given as

$$\dot{\hat{q}} = -a_m\hat{q} + \zeta(q,\ u) + a_m q + \hat{f}\left(q,\ u,\ \hat{\theta}\right) \tag{39}$$

where $\hat{f}\left(q,\ u,\ \hat{\theta}\right)$ is an online approximator. If the state-estimation error is defined as $\varepsilon = q - \hat{q}$, then the dynamic estimation error $\dot{\varepsilon}$ becomes

$$\dot{\varepsilon} = \dot{q} - \dot{\hat{q}} = -a_m\varepsilon + f(q,\ u) - \hat{f}\left(q,\ u,\ \hat{\theta}\right) \tag{40}$$

Based on the Universal Approximation Theorem for sigmoid neural networks, the fault can be represented by

$$\beta(t - T_0)f(q,u) = \theta^{*T}\sigma(q,u) + \varepsilon_o \tag{41}$$

where ε_o is a bounded approximation error.

Let $\tilde{\theta} = \theta^* - \hat{\theta}$ represent the SNN weight estimation error, and $Z = \frac{\partial \hat{f}}{\partial \hat{\theta}} = \sigma(q,u)$ represent the activation matrix. The estimation error dynamics reduce to

$$\dot{\varepsilon} = -a_m\varepsilon + \tilde{\theta}^T Z + \varepsilon_N \tag{42}$$

where ε_N combines the SNN approximation error and higher-order linearization residuals.

To establish stability for tracking and the dynamic estimator parameters, a positive-definite composite Lyapunov candidate function can be employed

$$V = \frac{1}{2}\varepsilon^T\varepsilon + \frac{1}{2}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta} \tag{43}$$

Differentiating V with respect to time yields

$$\dot{V} = \varepsilon^T\dot{\varepsilon} - \tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}} \tag{44}$$

$$\dot{V} = \varepsilon^T(-a_m\varepsilon + \tilde{\theta}^T Z + \varepsilon_N) - \tilde{\theta}^T\Gamma^{-1}\dot{\hat{\theta}} \tag{45}$$

$$\dot{V} = -a_m\|\varepsilon\|^2 + \varepsilon^T\varepsilon_N + \tilde{\theta}^T(Z^T\varepsilon - \Gamma^{-1}\dot{\hat{\theta}}) \tag{46}$$

Since the term $(Z^T\varepsilon - \Gamma^{-1}\dot{\hat{\theta}}) = 0$, we get

$$\dot{V} = -a_m\|\varepsilon\|^2 + \varepsilon^T\varepsilon_N \tag{47}$$

Next, applying Young's Inequality to the estimation bounding inner product yields

$$\varepsilon^T\varepsilon_N \le \frac{a_m}{2}\|\varepsilon\|^2 + \frac{1}{2a_m}\|\varepsilon_N\|^2 \tag{48}$$

Since neural-network approximation error ε_N and tracking error e are bounded, the term $\dot{V}$ is negative outside a compact residual set D that bounds the disturbances and friction terms. This guarantees that tracking error e, state-estimation error ε, and neural-network parameters $\hat{\theta}$ remain globally uniformly ultimately bounded.

Now, if the neural network eliminates the nonlinear fault and disturbance terms entirely, then

$$\dot{V} \leq -\frac{a_m}{2}\|\varepsilon\|^2 \leq 0 \tag{49}$$

Because $V \geq 0$ and $\dot{V} \leq 0$, the signals $\varepsilon, e \in L_2 \cap L_\infty$. Since their derivatives remain bounded, Barbalat's Lemma guarantees that both state-estimation error and tracking error converge asymptotically to zero, hence $\lim_{t\to\infty} \varepsilon(t) = 0$, $\lim_{t\to\infty} e(t) = 0$.

3.6. Design of Intelligent Control System

In this section, the intelligent control scheme is developed to detect the nonlinear fault and approximate it, so that the nominal controller can tolerate the fault and track the command signal accurately. The stability and tracking performance should also be maintained in the presence of modeling uncertainties.

Figure 4 shows the MATLAB/SIMULINK model of the fault-tolerant control system of the robotic manipulator. As shown in this Figure, the control system is made up of the following parts: the robotic manipulator block, nonlinear controller block, state-estimation block, threshold block, neural-network activation block, and dynamic identification block.

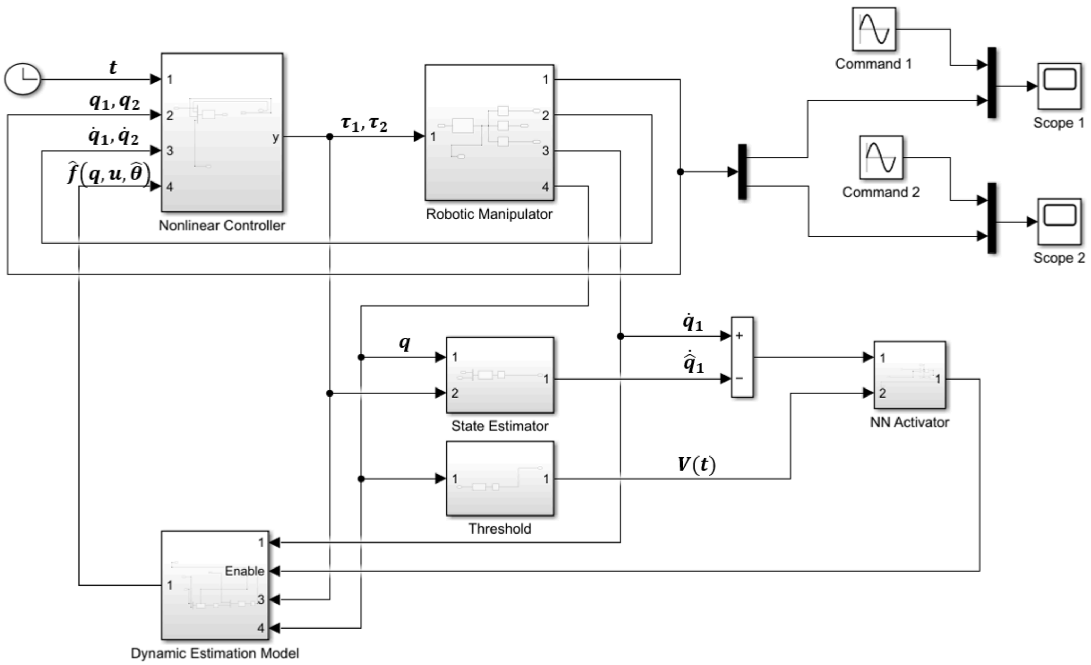


Figure 4. MATLAB/SIMULINK model of the fault-tolerant control system of a robotic manipulator.

The functions of the fault-tolerant control system parts in Figure 1 are summarized as follows:

1. Robotic manipulator block: It contains the nonlinear dynamics of the 2-link robotic manipulator with modeling uncertainties and nonlinear fault, where the system dynamics and the fault vector are defined in (14) and (15), respectively. It should be noted that the time occurrence of the nonlinear fault is unknown, which is subject to the step function defined in (20).

2. Nonlinear control block: It represents the neural-network-based nonlinear control law, which is defined by

$$\tau = M(q)\left(\ddot{q}_d + K_d\dot{e} + K_p e\right) + V_m(q,\ \dot{q})\dot{q} + G(q) - M(q)\hat{f}\left(q,\ u,\ \hat{\theta}\right). \tag{50}$$

In (50), it should be noted that the online approximator $\hat{f}\left(q,\ u,\ \hat{\theta}\right)$ is included in the nominal control law to obtain a robust nonlinear controller that compensates for the unknown fault based on the neural network.

3. State estimator block: It is used to estimate the state variables of the robotic manipulator. This block utilizes the adaptive laws that generate the estimated states, which are compared with the states of the robotic system dynamics to obtain a state-estimation error that oscillates within the limits of modeling uncertainties.
4. Threshold block: It contains the adaptive threshold that generates the threshold signal that detects the nonlinear fault. Once the fault occurs, the state-estimation error (or the residual of the state estimator) exceeds the modeling uncertainty limits and hits the threshold signal.
5. NN activator block: This block activates the sigmoid neural network in the dynamic identification block in case of fault occurrence. When the fault is detected by the adaptive threshold, the online approximator is enabled by the activation block to estimate the unknown fault.
6. Dynamic identification block: It contains the dynamics estimation model along with the adaptive laws and the online approximator. This block is enabled by the neural-network activator, so that the unknown fault signal is estimated and fed back to the nonlinear control block to be compensated.

The operation of the fault-tolerant control system is summarized as follows. Before fault occurrence, the fault signal $f(q,u)$ equals zero, and the adaptive threshold is greater than the residual signal; therefore, the robotic system is governed by the nominal control law.

After fault occurrence, $f(q,u)$ changes the system dynamics, and the nominal control law becomes unable to govern the system and track the desired trajectory. Thus, the residual signal becomes larger than the threshold. Once the fault is detected, the online approximator $\hat{f}\left(q,\ u,\ \hat{\theta}\right)$ is activated, and it starts to approximate the fault function and update the weights accordingly. If the weights reach the optimal values, then

$$\left|f(q,u) - \hat{f}\left(q,\ u,\ \hat{\theta}\right)\right| < \varepsilon \tag{51}$$

where ε is a very small value.

When the online approximator is included in the nominal control law, the latter becomes

$$\tau = M(q)\left(\ddot{q}_d + K_d\dot{e} + K_p e\right) + V_m(q,\ \dot{q})\dot{q} + G(q) - M(q)\hat{f}\left(q,\ u,\ \hat{\theta}\right) \tag{52}$$

Applying the controller given above to the robotic system, the closed-loop error dynamics yield

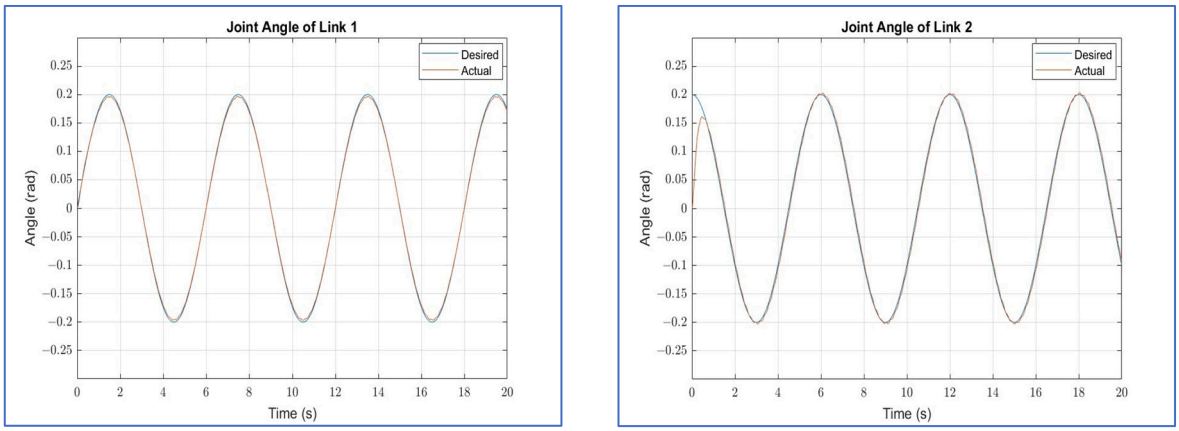
$$\ddot{e} + K_d\dot{e} + K_p e + M^{-1}(q)(-F(\dot{q}) - \tau_d) + f(q,u) - \hat{f}\left(q,\ u,\ \hat{\theta}\right) = 0 \tag{53}$$

Since the fault function and SNN output are close to each other, the difference between them becomes negligible. In addition, the gains K_d and K_p are selected such that the closed-loop poles of the system are located on the LHS of the s-plane. Therefore, the tracking error converges to a small value, and the state of the robotic system tracks the desired trajectory.

4. Simulation Results and Discussion

In this Section, the robust fault-tolerant control system of the 2-link robotic manipulator is constructed using MATLAB/SIMULINK in Figure 5. The robotic system model,

nonlinear controller, state estimator, adaptive threshold, and dynamic identification model are developed using MATLAB functions. The robotic manipulator parameters are given in Section 2, while the modeling uncertainty parameters θ_1 and θ_2 are assumed to be 0.45 and 0.4, respectively.



(**a**) (**b**)

Figure 5. Simulation results of the joint angles of (**a**) link 1 and (**b**) link 2 in the intelligent nonlinear controlled robotic system under modeling uncertainties.

Off-line training has been conducted to obtain the optimal initial conditions, which are:

$$x_0 = [0.2559 \quad -0.8233 \quad 0.2372 \quad 0.2896 \quad 0.2471 \quad -0.9538$$
$$-0.2302 \quad 0.7743 \quad 0.2982 \quad -0.2005 \quad 0.2170 \quad -0.4236 \quad -0.1688$$
$$0.5006 \quad 0.4091 \quad -0.7681 \quad 0.3437 \quad -0.2677 \quad 0.9211 \quad -0.7818$$
$$-0.4162 \quad 0.5217 \quad -0.6944 \quad 0.0518 \quad 0.9751 \quad 0.8885 \quad -0.3574$$
$$-0.2459 \quad -0.4708 \quad -0.0752 \quad 0.2860 \quad -1.2349 \quad 1.0241 \quad 0.3398$$
$$-0.8301 \quad 0.1713 \quad 0.3038 \quad 0.0986 \quad -0.1600 \quad -0.6152 \quad -0.5429$$
$$0.1578 \quad -0.5737 \quad -0.4092 \quad 0.7174 \quad 0.6520 \quad 0.5065 \quad 0.7411$$
$$0.0080 \quad 0.2017 \quad 0.2226 \quad -0.0186].$$

4.1. Before Fault Occurrence

The simulation is conducted with modeling uncertainty and no-fault effect. The purpose of this part is to make sure that the intelligent nonlinear controller governs the system dynamics. Additionally, the state-estimation error should be bounded, and the threshold should not produce any false alarms due to modeling uncertainty. Figure 5 shows the desired (command) and actual joint angle responses of links 1 and 2, respectively. The control effort is shown in Figure 6. It can be noticed that the robotic system tracks the desired trajectory properly in the presence of modeling uncertainties.

Figure 7 shows the residual (state-estimation error) and the threshold signals of the fault-tolerant control system of the robotic manipulator. It can be noticed that there is a bounded residual signal around zero due to the modeling uncertainty. It can also be noticed that the threshold signal and the residual signal do not intersect if there is no fault in the system. In other words, the intelligent controller does not produce any false alarms during normal operating conditions.

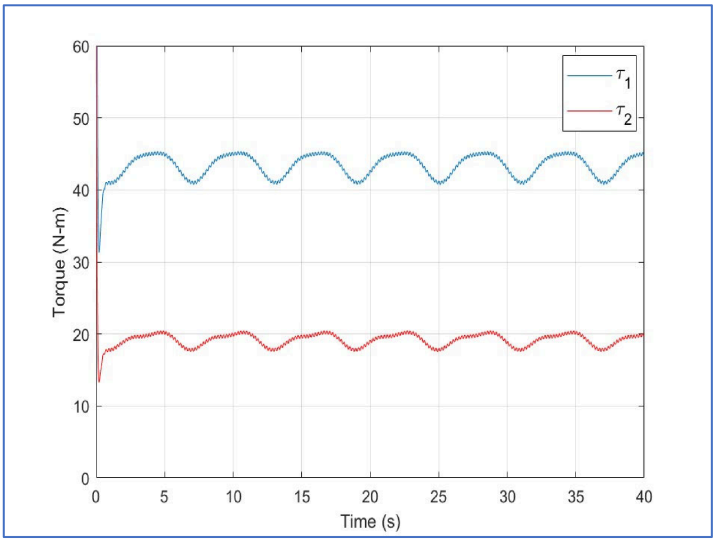


Figure 6. Control signals of the intelligent nonlinear controlled robotic system under normal operation with modeling uncertainties.

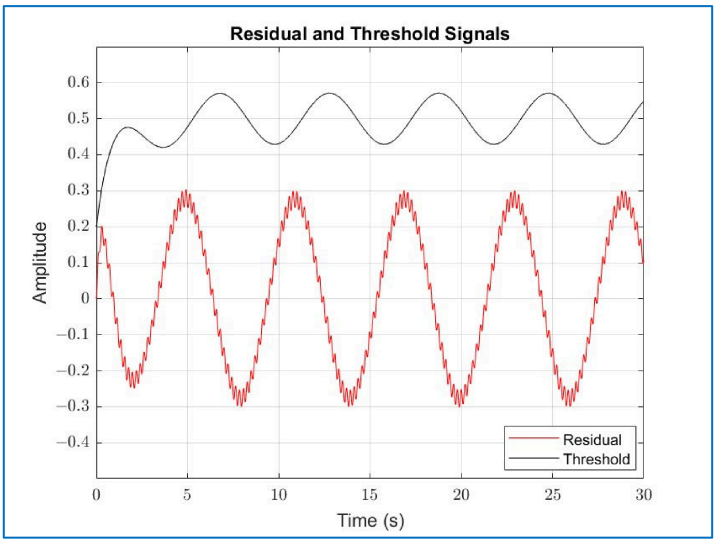


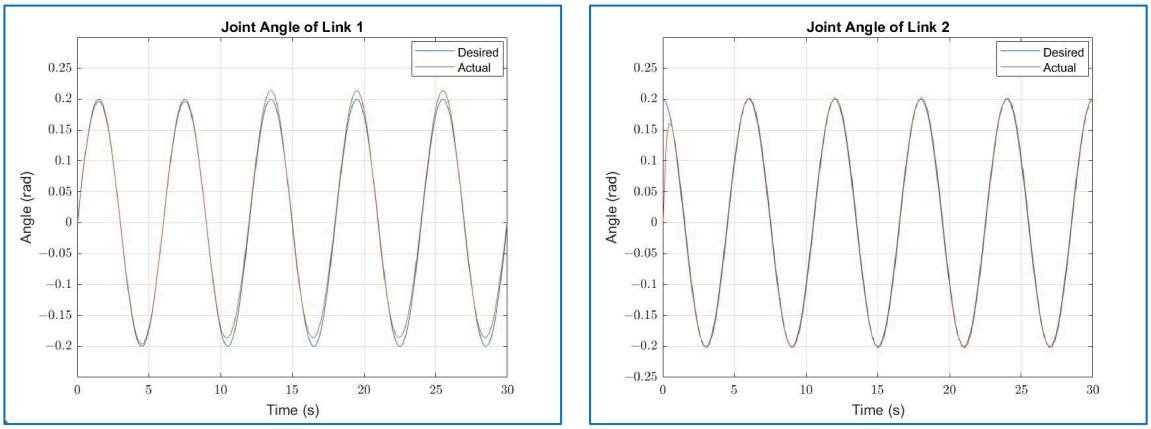
Figure 7. Threshold and residual signals under normal operation.

4.2. After Fault Occurrence

4.2.1. Fault Detection

In this case, a MATLAB simulation has been conducted for the fault-tolerant control system of the robotic manipulator with modeling uncertainty, while the dynamic identification model and the online approximator are disabled. The nonlinear fault defined in (15) is applied to the robotic system dynamics at time $t \geq 10$ s.

Figure 8 shows the desired and actual joint angle responses of links 1 and 2, respectively. It can be noticed that the actual joint angle response of link 1 (q_1) deviates from the desired trajectory at $t \geq 10$ s. However, the actual joint angle response of link 2 (q_2) still tracks the desired trajectory accurately. This is because of the nonlinear fault that is injected into the system dynamics at $\dot{q}_1$, which affects the joint angle response of link 1 only.



(**a**) (**b**)

Figure 8. Desired and actual joint angle response of (**a**) link 1 and (**b**) link 2 of the intelligent nonlinear controlled robotic system under modeling uncertainties and nonlinear fault.

Based on Figure 8, it can be noticed that the nominal controller does not support the robotic system in tracking the desired trajectory in the presence of a fault. In other words, the tracking error of q_1 response increases after $t = 10$ s due to the fault effect in $\dot{q}_1$. Hence, in order to improve the tracking performance of the nominal controller, it should be upgraded to an intelligent controller that tracks the desired trajectory under nonlinear faults. The intelligence property of the fault-tolerant controller is inherited from the capability of the detection, approximation, and compensation of nonlinear faults in order to maintain a precise tracking performance.

To investigate the detection capability of the control system under nonlinear fault conditions and modeling uncertainty, the dynamic identification model is activated during the simulation. Now, the adaptive threshold should detect the nonlinear fault within a reasonable time frame. Figure 9 shows the angular velocity residual signal and the adaptive threshold signal of the intelligent control system of the robotic manipulator under fault conditions, which is injected at $t \geq 10$ s. It can be noticed that the residual signal goes above the bounding limits of the modeling uncertainty when the fault occurs at $t = 10$ s. Obviously, the threshold signal intersects with the residual signal at $t = 10.3$ s, and the fault detection is achieved successfully.

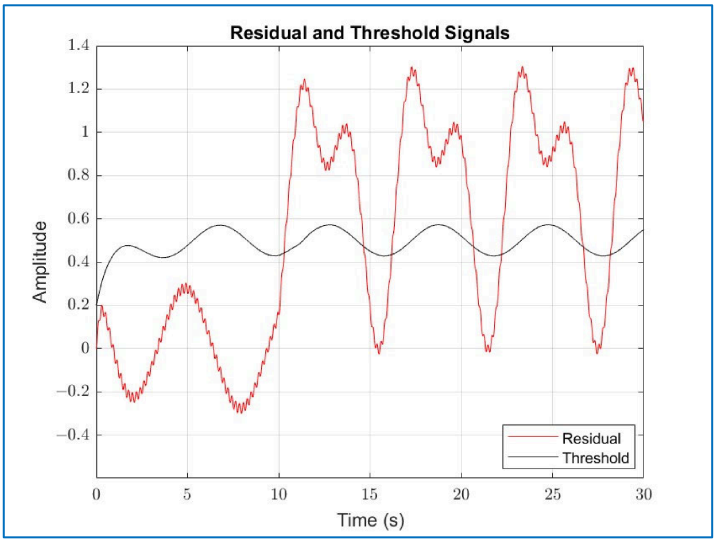


Figure 9. Threshold and residual signals under nonlinear fault condition.

4.2.2. Fault Accommodation

Once the fault is detected, a signal is sent to the neural-network activator in order to activate the dynamic identification model. In this model, the adaptive laws adjust the neural-network weights online, so that the neural network learns the nonlinear fault function based on the updated weights. Then, a feedback signal is sent to the nominal controller to compensate for the major part of the fault function and provide accurate tracking performance. Thus, the tracking error converges to a value that is close to zero, and the stability is maintained.

Figure 10 exhibits the time response of the neural-network output and the residual signal. The time response of the neural-network approximation error is also plotted in Figure 11. It can be noticed that the neural network tries to approximate the nonlinear fault, but the approximation performance is not perfect due to the system complexity, which has 4 states. Nevertheless, the neural-network approximation error is small, as shown in Figure 11, which means that the approximation performance of the neural network is acceptable.

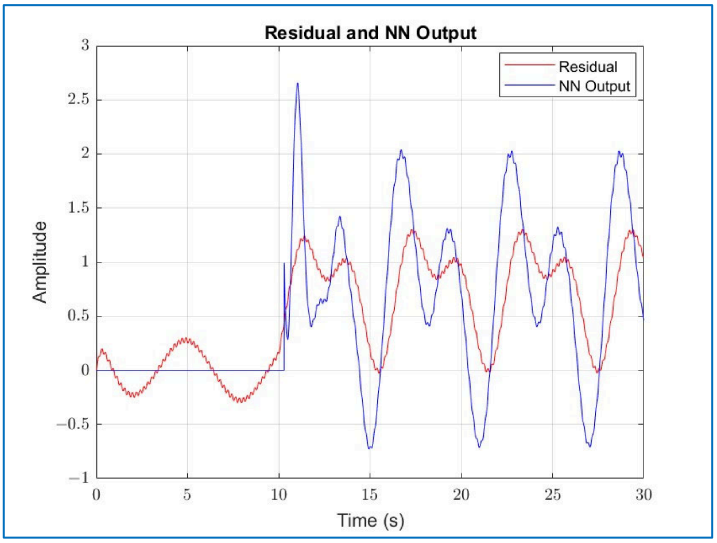


Figure 10. The time response of the residual signal and the neural-network output.

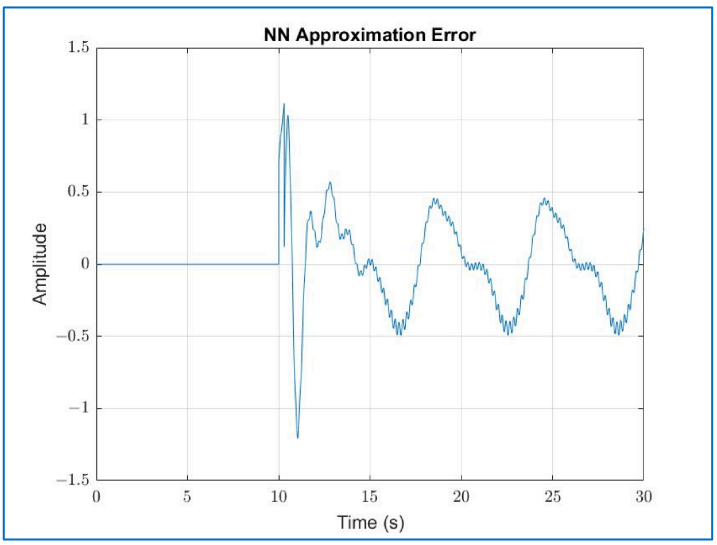
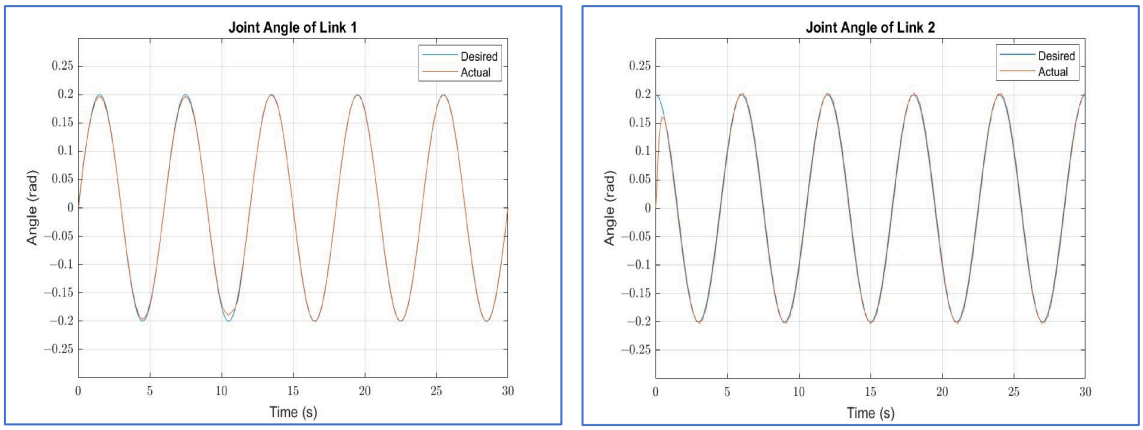


Figure 11. Neural-network approximation error.

Figures 12a and 12b show the desired and actual joint angle response of links 1 and 2, respectively. It can be seen that when the fault is detected and accommodated after $t = 10$ s, the joint angles can track the desired trajectories properly. The control effort is shown in Figure 13, while the tracking error of the joint angles of links 1 and 2 is shown in Figures 14a and 14b, respectively. It can be observed that in the presence of modeling uncertainties and nonlinear faults, the joint angle tracking error of link 1 is minimized compared to that in the nominal control case. The response in Figure 14 also indicates that the NN eliminates the nonlinear fault and the major part of the modeling uncertainty.



(**a**) (**b**)

Figure 12. Desired and actual joint angle response of (**a**) link 1 and (**b**) link 2 of the intelligent nonlinear controlled robotic system during fault detection and accommodation.

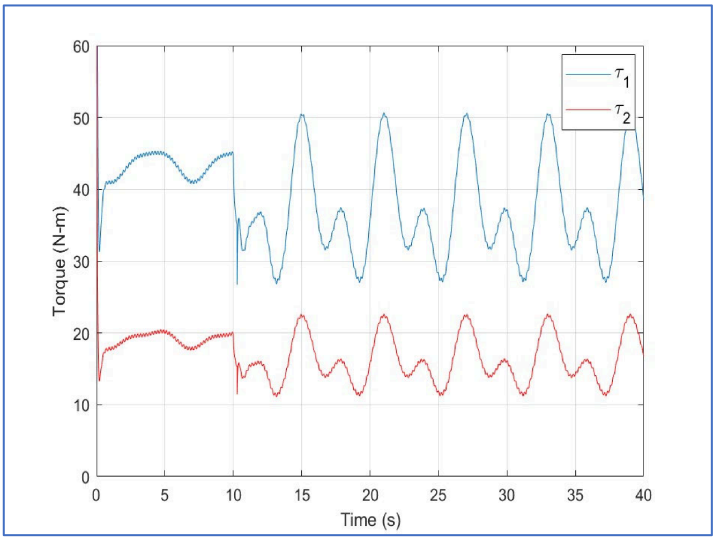
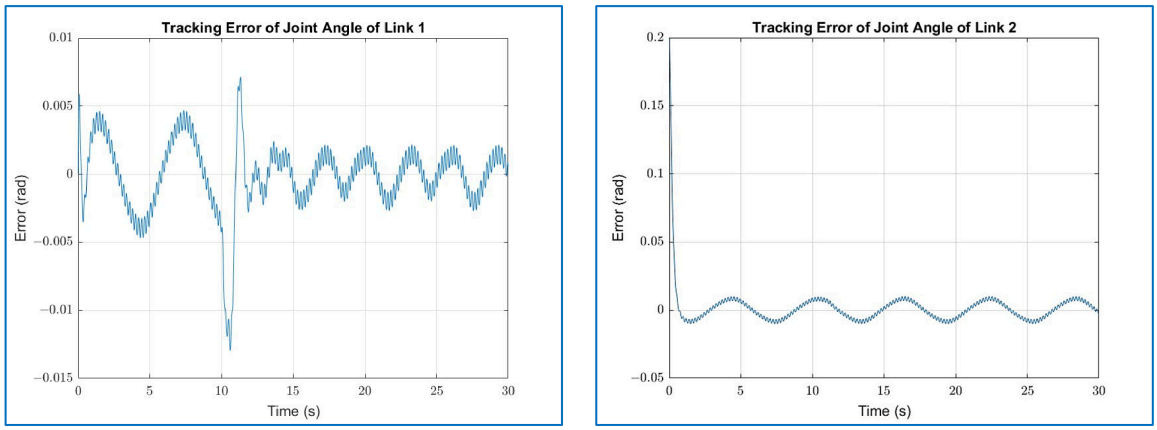


Figure 13. Control signals of the intelligent nonlinear controlled robotic system under nonlinear fault and modeling uncertainties.

Table 2 shows performance indices of the proposed FTC system. Clearly, RMSE and maximum tracking error of the robotic system position response are reduced when the FTC scheme is activated after the fault occurrence. In addition, the fault recovery time and fault detection delay of the FTC system are 1 s and 300 ms, respectively. It can also be seen that the NN approximation RMSE of the proposed control scheme is reduced from 0.799 to 0.278.



(**a**) (**b**)

Figure 14. Joint angle tracking error of (**a**) link 1 and (**b**) link 2 of the intelligent nonlinear controlled robotic system during fault detection and accommodation.

Table 2. Quantitative performance indices for the proposed FTC system.

Performance Index	RMSE	Maximum Tracking Error	Fault Detection Delay	Fault Recovery Time	NN Approximation RMSE
Healthy nominal case	0.0028	0.006	—	—	0
Faulty case	0.0087	0.017	—	—	0.799
Proposed FTC case	0.0024	0.013	300 ms	1.0 s	0.278

5. Conclusions

The tracking performance of the nominal control system of the 2-link robotic manipulator has been investigated under modeling uncertainties and normal operating conditions. It has been shown that the tracking error converges to a small value rather than zero due to the modeling uncertainties. However, under nonlinear fault conditions, the tracking performance of the nominal control system is deteriorated, so an intelligent controller is needed to accommodate the fault effect and track the desired trajectory.

The design approach of the robust fault-tolerant control scheme has been provided in detail. The main parts of the control system include an estimator to estimate the states under modeling uncertainties, an adaptive threshold to detect the fault, and a dynamic estimation model to approximate the unknown fault. The stability analysis of the closed-loop control system is provided to ensure that the tracking error converges to a small value that is bounded by the upper limit of modeling uncertainties. The adaptive threshold of the fault-tolerant control scheme should satisfy two main conditions. First, it should be designed such that no false alarm is produced due to modeling uncertainties. Second, it should detect the fault within a reasonable time frame. The simulation results have shown that the fault is detected within 300 ms. Moreover, no false alarm has been produced by the adaptive threshold during normal operating conditions.

The learning capability of the online approximator using a neural network has been verified. It has been noticed that the dynamic estimation model is capable of adjusting the neural-network weights, so that the major part of the fault is approximated efficiently. It has also been confirmed that the tracking performance of the nominal control system, which has been designed based on the computed-torque method, has been improved significantly with the incorporation of the proposed fault-tolerant control technique.

Author Contributions: Conceptualization, H.A.-B.; methodology, H.A.-B.; software, H.A.-B.; validation, H.A.-B.; formal analysis, H.A.-B.; resources, M.K.K. and A.A.E.; visualization, A.A.E.; writing—original draft preparation, H.A.-B.; writing—review and editing, M.K.K.; supervision, M.K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Zhang, J.; Song, J.-G.; Chen, Q.; Yang, F. Mixed-Gain Adaption-Based Fault-Tolerant Funnel Control of Robotic Manipulators With Unknown Dynamics and Sensor Faults. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 3044–3055. [CrossRef]
2. Lochan, K.; Seneviratne, L.; Hussain, I. Adaptive Global Super-Twisting Sliding Mode Control for Trajectory Tracking of Two-Link Flexible Manipulators. *IEEE Access* **2025**, *13*, 61876–61889. [CrossRef]
3. Urrea, C.; Kern, J.; Torres, V. Design, Simulation, and Comparison of Advanced Control Strategies for a 3-Degree-of-Freedom Robot. *Appl. Sci.* **2024**, *14*, 11010. [CrossRef]
4. Zhang, J.-X.; Qi, W.-L.; Chai, T. Constrained Tracking Control of MIMO Nonlinear Systems With Discontinuous References and Unknown Dynamics. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 15139–15149. [CrossRef]
5. Cheng, H.; Song, Y. Performance Guaranteed Robust Tracking Control of MIMO Nonlinear Systems With Input Delays: A Global and Low-Complexity Solution. *IEEE Trans. Cybern.* **2024**, *54*, 7793–7803. [CrossRef] [PubMed]
6. Liu, H.; Huang, Q.; Liu, X.; Zhen, S. Robust Approximate Constraint-Following Control Design for Two-Link Robotic Manipulator System. In Proceedings of the 2023 IEEE International Conference on Development and Learning (ICDL), Macau, China, 9–11 November 2023; pp. 200–205.
7. Humaloja, J.-P.; Koivumäki, J.; Paunonen, L.; Mattila, J. Decentralized Observer Design for Virtual Decomposition Control. *IEEE Trans. Autom. Control* **2022**, *67*, 2529–2536. [CrossRef]
8. Ma, Y.; Zhao, H.; Li, T. Adaptive Super-Twisting Tracking for Uncertain Robot Manipulators Based on the Event-Triggered Algorithm. *Sensors* **2025**, *25*, 1616. [CrossRef] [PubMed]
9. Dao, H.V.; Nguyen, M.H.; Ahn, K.K. Nonlinear Functional Observer Design for Robot Manipulators. *Mathematics* **2023**, *11*, 4033. [CrossRef]
10. Massaro, M.; Lovato, S.; Bottin, M.; Rosati, G. An Optimal Control Approach to the Minimum-Time Trajectory Planning of Robotic Manipulators. *Robotics* **2023**, *12*, 64. [CrossRef]
11. Chunqing, H.; Songjiao, S. PID feedback for mixed H2/H∞ tracking control of robotic manipulators. *J. Syst. Eng. Electron.* **2004**, *15*, 579–585.
12. Liu, G.; Wang, Y.; Li, J.; Wang, Q. Iterative Learning Control with Forgetting Factor for MIMO Nonlinear Systems with Randomly Varying Iteration Lengths and Disturbances. *Symmetry* **2025**, *17*, 694. [CrossRef]
13. Eltayeb, A.; Ahmed, G.; Imran, I.H.; Alyazidi, N.M.; Abubaker, A. Comparative Analysis: Fractional PID vs. PID Controllers for Robotic Arm Using Genetic Algorithm Optimization. *Automation* **2024**, *5*, 230–245. [CrossRef]
14. Mohamed, M.J.; Oleiwi, B.K.; Abood, L.H.; Azar, A.T.; Hameed, I.A. Neural Fractional Order PID Controllers Design for 2-Link Rigid Robot Manipulator. *Fractal Fract.* **2023**, *7*, 693. [CrossRef]
15. Al-Baidhani, H.; Kazimierczuk, M.K. Design and Implementation of Digital PID Control for Mass-Damper Rectilinear Systems. *Mathematics* **2024**, *12*, 2921. [CrossRef]
16. Kim, Y.H.; Lewis, F.L. Neural network output feedback control of robot manipulators. *IEEE Trans. Robot. Autom.* **1999**, *15*, 301–309. [CrossRef]
17. Vlachos, C.; Tolis, F.; Karras, G.C.; Bechlioulis, C.P. Neural Network Iterative Learning for SISO Non-Affine Control Systems. *Electronics* **2024**, *13*, 1473. [CrossRef]
18. Xiao, L.; Li, S.; Li, K.; Jin, L.; Liao, B. Co-Design of Finite-Time Convergence and Noise Suppression: A Unified Neural Model for Time Varying Linear Equations With Robotic Applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 5233–5243. [CrossRef]
19. Sun, Y.; Liang, X.; Wan, Y. Tracking Control of Robot Manipulator with Friction Compensation Using Time-Delay Control and an Adaptive Fuzzy Logic System. *Actuators* **2023**, *12*, 184. [CrossRef]

20. Sun, J.; Wu, S.; Chen, J.; Li, X.; Wu, Z.; Xia, R.; Pan, W.; Zhang, Y. Optimization of Fuzzy Adaptive Logic Controller for Robot Manipulators Using Modified Greater Cane Rat Algorithm. *Mathematics* **2025**, *13*, 1631. [CrossRef]
21. Xu, W.; Yang, L.; Lai, G.; Chen, Y. Finite-Time Adaptive Visual Tracking Control of Manipulators With Parameter Uncertainties. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 4976–4986. [CrossRef]
22. Vaishnavi, J.; Singh, B.; Kumar, R. Design and Analysis of Trajectory Tracking Controllers for Noisy 2-Link Robotic Manipulator. In Proceedings of the 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), Mysuru, India, 10–11 December 2021; pp. 544–549.
23. Guo, K.; Shi, P.; Wang, P.; He, C.; Zhang, H. Non-Singular Terminal Sliding Mode Controller with Nonlinear Disturbance Observer for Robotic Manipulator. *Electronics* **2023**, *12*, 849. [CrossRef]
24. Zhang, D.; Hu, J.; Cheng, J.; Wu, Z.-G.; Yan, H. A Novel Disturbance Observer Based Fixed-Time Sliding Mode Control for Robotic Manipulators with Global Fast Convergence. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 661–672. [CrossRef]
25. Zhu, Q.; Zhang, J.; Liu, Z.; Yu, S. Robust Higher-Order Nonsingular Terminal Sliding Mode Control of Unknown Nonlinear Dynamic Systems. *Mathematics* **2025**, *13*, 1559. [CrossRef]
26. Guo, K.; Wei, C.; Shi, P. Fuzzy Disturbance Observer-Based Adaptive Nonsingular Terminal Sliding Mode Control for Multi-Joint Robotic Manipulators. *Processes* **2025**, *13*, 1667. [CrossRef]
27. Duong, T.T.C.; Nguyen, C.C.; Tran, T.D. Experimental Investigation of Motion Control of a Closed-Kinematic Chain Robot Manipulator Using Synchronization Sliding Mode Method with Time Delay Estimation. *Appl. Sci.* **2025**, *15*, 5206. [CrossRef]
28. Song, Z.; Bao, D.; Wang, W.; Zhao, W. Adaptive Dynamic Boundary Sliding Mode Control for Robotic Manipulators under Varying Disturbances. *Electronics* **2024**, *13*, 900. [CrossRef]
29. Rahmani, M.; Redkar, S. Robot Manipulator Control Using a Robust Data-Driven Method. *Fractal Fract.* **2023**, *7*, 692. [CrossRef]
30. Al-Baidhani, H.; Kazimierczuk, M.K.; Ordóñez, R. Nonlinear Modelling and Control of PWM DC-DC Buck-Boost Converter for CCM. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 1374–1379.
31. Al-Baidhani, H.; Kazimierczuk, M.K. Simplified Double-Integral Sliding-Mode Control of PWM DC-AC Converter with Constant Switching Frequency. *Appl. Sci.* **2022**, *12*, 10312. [CrossRef]
32. Domínguez, C.; Urrea, C. Intelligent Fault-Tolerant Control of Delta Robots: A Hybrid Optimization Approach for Enhanced Trajectory Tracking. *Sensors* **2025**, *25*, 1940. [CrossRef] [PubMed]
33. Milecki, A.; Nowak, P. Review of Fault-Tolerant Control Systems Used in Robotic Manipulators. *Appl. Sci.* **2023**, *13*, 2675. [CrossRef]
34. Zhang, Y.; Kong, L.; He, W.; Knoll, A. Fixed-Time Prescribed Performance Neural Fault-Tolerant Control of Euler–Lagrange Systems Under Unknown Bounded Initial Conditions. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2026**, *73*, 1317–1330. [CrossRef]
35. Su, J.; Song, Y. Fault-Tolerant Control Achieving Prescribed Tracking Accuracy Within Given Time for Euler-Lagrange Systems Under Unknown Actuation Characteristics and Fading Powering Faults. *IEEE/CAA J. Autom. Sin.* **2026**, *13*, 72–82. [CrossRef]
36. Xu, K.-D.; Zhang, J.-X.; Chai, T. Fault-Tolerant Prescribed Performance Control of Cascade Nonlinear Systems With Sensor Faults and Unknown Dynamics. *IEEE Trans. Ind. Electron.* **2026**, *73*, 2926–2935. [CrossRef]
37. Zhang, G.; Cheng, D. Adaptive Fault-Tolerant Guaranteed Performance Control for Euler-Lagrange Systems With Its Application to a 2-Link Robotic Manipulator. *IEEE Access* **2020**, *8*, 184160–184171. [CrossRef]
38. Sun, D.; Zhang, N.; An, T. Adaptive Nonsingular Predefined-Time Sliding Mode Fault Tolerant Control for Manipulator System With Given Steady-State Error. *IEEE Access* **2026**, *14*, 31596–31605. [CrossRef]
39. Shan, H.; Jiang, Y.; Zhu, Y.; Wang, S.; Rubio, J.d.J. Dynamic-Memory Event-Triggered Fuzzy Adaptive Fault-Tolerant Control for Robotic Manipulators: A New Switching Mechanism. *IEEE Trans. Fuzzy Syst.* **2026**, *34*, 152–162. [CrossRef]
40. Huang, H.; Yang, Y.; Chai, W.; Li, Z.; Zhou, H. A Knowledge Intelligent Fault-Tolerant Soft Robotic Operation Framework for Alterable Grasp Configuration Soft Gripper. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 379–389. [CrossRef]
41. Xia, Y.; Yuan, Y.; Sun, W. Finite-Time Adaptive Fault-Tolerant Control for Robot Manipulators With Guaranteed Transient Performance. *IEEE Trans. Ind. Inform.* **2025**, *21*, 3336–3345. [CrossRef]
42. Wang, D.; Zhang, G.; Chen, R.; Zhang, J.; Zhang, T. Time-Synchronized Fault-Tolerant Control for Robotic Manipulators. *Mathematics* **2025**, *13*, 507. [CrossRef]
43. Tran, X.-T.; Nguyen, V.-C.; Le, P.-N.; Kang, H.-J. Observer-Based Fault-Tolerant Control for Uncertain Robot Manipulators without Velocity Measurements. *Actuators* **2024**, *13*, 207. [CrossRef]
44. Roshanravan, S.; Shamaghdari, S. Integrated Fault Detection and Fault-Tolerant Optimal Tracking Control for Unknown Nonlinear Systems With State and Input Constraints Using Safe Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2026**, *56*, 46–58. [CrossRef]
45. 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]

46. Zhang, L.; Hou, Y.; Liu, H.; Tang, D.; Li, L. Prescribed Performance Fault-Tolerant Tracking Control of Uncertain Robot Manipulators with Integral Sliding Mode. *Mathematics* **2023**, *11*, 2430. [CrossRef]
47. Ahmed, S.; Azar, A.T.; Tounsi, M. Adaptive Fault Tolerant Non-Singular Sliding Mode Control for Robotic Manipulators Based on Fixed-Time Control Law. *Actuators* **2022**, *11*, 353. [CrossRef]
48. Vemuri, A.T.; Polycarpou, M.M.; Diakourtis, S.A. Neural network based fault detection in robotic manipulators. *IEEE Trans. Robot. Autom.* **1998**, *14*, 342–348. [CrossRef]
49. You, X.; Dian, S.; Guo, B.; Xiao, Q.; Zhu, Y.; Liu, K. Novel Finite-Time Adaptive Fuzzy Fault-Tolerant Control for Fractional-Order Nonlinear Systems and Its Applications. *IEEE/CAA J. Autom. Sin.* **2026**, *13*, 123–136. [CrossRef]
50. Yue, S.; Zhang, L.; Zhao, N.; Xu, N.; Alsaadi, F.E. SMS-Based Event-Triggered Fault Tolerant Control of Nonlinear Systems Using Adaptive Dynamic Programming. *Int. J. Robust Nonlinear Control* **2026**, *36*, 3752–3776. [CrossRef]
51. El Fezazi, Y.; El Fezazi, N.; Idrissi, S.; Houssaine Tissir, E. Robust Fault Estimation and Fault-Tolerant Control in T–S Fuzzy Systems Using $H\infty$ Optimization: Application to Lateral Vehicle Dynamics. *Int. J. Adapt. Control Signal Process.* **2026**, *40*, 228–251. [CrossRef]
52. Lewis, F.L.; Jagannathan, S.; Yesildirek, A. *Neural Network Control of Robot Manipulators and Nonlinear Systems*; Taylor Francis: Padstow, UK, 1999.

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.