

Review

# Review of Fault-Tolerant Control Systems Used in Robotic Manipulators

Andrzej Milecki \*  and Patryk Nowak 

Division of Mechatronic Devices, Poznan University of Technology, 60-965 Poznan, Poland

\* Correspondence: [andrzej.milecki@put.poznan.pl](mailto:andrzej.milecki@put.poznan.pl)

**Abstract:** Control systems that ensure robot operation during failures are necessary, particularly when manipulators are operating in hazardous or hard-to-reach environments. In such applications, fault-tolerant robot controllers should detect failures and, using fault-tolerant control methods, be able to continue operation without human intervention. Fault-tolerant control (FTC) is becoming increasingly important in all industries, including production lines in which modern robotic manipulators are used. The use of fault-tolerant systems in robotics can prevent the production line from being immobilized due to minor faults. In this paper, an overview of the current state-of-the-art methods of fault-tolerant control in robotic manipulators is provided. This review covers publications from 2003 to 2022. The article pays special attention to the use of artificial intelligence (AI), i.e., fuzzy logic and artificial neural networks, as well as sliding mode and other control methods, in the FTC of robotic manipulators. The cited and described publications were mostly found using Google Scholar.

**Keywords:** fault-tolerant control; FTC; robot manipulators; artificial intelligence



**Citation:** Milecki, A.; Nowak, P. Review of Fault-Tolerant Control Systems Used in Robotic Manipulators. *Appl. Sci.* **2023**, *13*, 2675. <https://doi.org/10.3390/app13042675>

Academic Editors: Sheng Du, Xiongbo Wan, Wei Wang and Hao Fu

Received: 29 December 2022

Revised: 14 February 2023

Accepted: 16 February 2023

Published: 19 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The basic concepts and techniques of fault tolerance in computer systems were described in [1], which presented the different error classifications and the application of redundancy techniques to ensure the reliable operation of computers. The modeling and prediction of faults were also described, along with examples of fault-tolerant computers. In the early 1990s, Stengel [2], Veillette [3], and Patton [4] published the results of different applications of FTC systems. In [5], state-of-the-art methods for fault-tolerant control applications up until 1997 were presented.

In a fault-tolerant control system, it is necessary to distinguish concepts such as fault definition and fault tolerance. A fault is defined as the deviation of a parameter from an acceptable value. Fault tolerance is the ability of the system to continue operation regardless of faults. The origins of FTC systems are related to the need to use special control in critical safety devices and equipment. Examples of such devices are aircraft or nuclear power plants. In [6], which was published in 1990, one of the first applications of fault-tolerant control in robot manipulators was described and the measure of joint failure influences on the remaining dexterity of a kinematically redundant manipulator was quantified. This measure was used as a criterion and technique for calculating optimal configurations of fault-tolerant redundant manipulators. An example of a three-dimensional robot was shown that used a fourth actuator, which guaranteed half of the original dexterity, after the joint failure.

In the same period, i.e., in the 1990s, extensive research was undertaken to develop FTC systems for applications in manipulators intended for space research [7–9]. The foundations for the design of fault-tolerant manipulators were presented in [10]. In this article, the modeling tools and reliable evaluation methods of robots were described. The design aspects, together with a reliability evaluation, were presented. Based on the performed analyses, a methodology for designing fault-tolerant manipulators was proposed. Particular

attention was paid to the development of recommendations for the design of the arm with 10 degrees of freedom. It was proposed to use two actuators on each joint.

Techniques for a robust residual generation have been proposed by several authors. In order to ensure the successful operation of autonomous robots in remote or hazardous environments, fault tolerance is essential. In [11], different fault tolerance algorithms were proposed, where the possible failures of the robot components and the interdependence of these failures must be determined first. To achieve this, fault tree analysis can be a very useful tool. The tree structures developed using this type of analysis make it possible to both determine a block diagram of possible occurrences of robot faults and define the cause-and-effect relationships between failures.

In [12], a control method was proposed in which a model reference algorithm and a proportional integral derivative controller were applied in the fault-tolerant operation of robotic manipulators. This method was tested in a simulation that showed the effectiveness of the recovery algorithm, which enabled it to continue the movement on the assumed trajectory and reach the given end effect.

In 1987, Selkäinen and Halme [13] proposed the use of artificial intelligence, i.e., a real-time expert system, in a fault control solution, which was used in an algorithm to supervise fault detection and localization. In the case of sensor damage, the system automatically replaced the faulty measurement with an updated predictor model output signal. The investigated system was successfully tested in a pilot process using artificial intelligence to overcome physical faults.

The authors of [14] proposed the use of intelligent fault-tolerant control in a flexible assembly cell, where the fault-tolerant controller would immediately recognize errors and react in real time to such situations. In the described solution, the combination of an advanced autonomous supervision system was combined with a sensor-guided action generator. Experimental tests were conducted using a mobile two-arm robot system.

Groom et al. [15] described the real-time fault-tolerant control developed for a kinematically redundant manipulator. The authors used a fault-tolerance measure to enable the end effector of the manipulator to continuously follow a given trajectory. Finally, an algorithm was developed and used in real time for the control of a seven-degree-of-freedom commercially available manipulator.

The purpose of using FTC systems is to assure reliability, which should be maintained despite the occurrence of faults of an acceptable nature. Intensive work is currently underway to develop this type of control in equipment used during manufacturing processes. One of the advantages of using this type of control in the production process is not only the safety of critical equipment but also cost reduction. The reduction in production costs is influenced by the continued operation of the devices used. At the time of an acceptable FTC failure, there is no need to stop production for maintenance purposes. This results in a reduction in the costs associated with service calls and also saves time by preventing a halt in the production process.

Shin and Lee [16] presented a robust fault-tolerant control system, which can be used to overcome the failure of the robot manipulator actuators. The controller uses the algorithms for normal control (non-failed), fault detection, and control in case of faults to achieve the assumed task completion.

Fault-tolerant control systems can be divided into three main categories: passive FTCS (PFTCS), active FTCS (AFTCS), and hybrid FTCS (HFTCS). PFTCS are characterized by the fact that they operate offline and are only able to adapt to faults that are defined at the design stage. Passive fault-tolerant steer systems do not require fault detection and isolation and are less computationally complex than active systems. AFTCS consist of a reconfiguration mechanism and a fault detection identification (FDI) mechanism. The FDI subsystem provides information to the controller about the fault and the controller responds by reconfiguring itself or the controlled device. The FDI subsystem is a key component of an active FTCS as it detects the fault and immediately sends information to the controller. The immediate reconfiguration of the controller upon receiving information from the FDI

allows this method to be called active. There are also hybrid HFTCS controllers, which are a combination of PFTCS and AFTCS.

In this paper, all the above-mentioned methods used in applications for robotic arm steering are reviewed. The methodology for selecting articles for this paper was limited to the time frame of the proposed solutions, that is, mainly the last 20 years. In addition, it was decided to place the main emphasis only on solutions that use artificial intelligence and sliding-mode controllers. Furthermore, all the selected papers deal with the use of FTC methods in robot control. In addition, the papers that focus on the control of manipulators with up to 7 degrees of freedom were chosen for our review. Articles about other robots such as hexapods or mobile robots were excluded from this review. This review is presented in six chapters, covering the available solutions of various FTC systems used to control robotic arms. In Section 1, the history and background of FTC systems are described. Section 2 is devoted to presenting the state-of-the-art FTCS that use artificial intelligence methods. Section 3 presents the variants of FTCS that use sliding-mode control. Section 4 deals with other FTCS used in robots that have been developed in recent years. Section 5 presents the summary and Section 6 presents the conclusions.

## 2. Artificial Intelligence Methods in FTCS

Artificial intelligence methods are highly developed techniques that have applications in fault-tolerant strategies for controlling robotic arms. Among the artificial intelligence methods used for fault-tolerant control systems are fuzzy logic controllers, which are able to effectively deal with non-linear systems using membership functions, which assess analog input signals using logical variables that take values between 0 and 1. The authors of [17] described the so-called adaptive fuzzy type-2 backstepping control (AFT2BC) method. The backstepping method was also described in [18]. AFT2BC was tested on a model of a PUMA560 robotic arm in a MATLAB environment in which the kinematic configuration of the robot could be changed simultaneously to simulate axis failure and changes in the load on the arm. The proposed control algorithm did not require a priori knowledge of the dynamic robot model so the controller could operate when the faulty condition was due to both model uncertainty and external disturbances. Figure 1 shows a block scheme of the proposed adaptive control method. On the description of the graph is a set of Lyapunov Equations (1)–(6) [17]:

$$\dot{V}_1 = e_1 \dot{e}_1 = e_1(\dot{q}_d - x_2), \quad (1)$$

where  $\dot{V}_1$  is the derivative of the Lyapunov function [19],  $e_1$  is the tracking error,  $\dot{q}_d$  is the derivative of the desired signal,  $x_2$  is the vector of the current state  $x_2 = \dot{x}_1$ ,  $x_1 = [q_1 q_2 q_3]^T$ ,

$$u_a = W^T(e_1 \dot{e}_1) \Theta, \quad (2)$$

where  $u_a$  is the fuzzy type-2 adaptive control expression that is designed to estimate the ideal backstepping control law,  $W^T$  represents the average basis functions obtained by a fuzzy type-2 system, where each basis function is given by the average of the corresponding left and right basis functions,  $\Theta$  denotes the adapted vector parameters,

$$\dot{\Theta} = \gamma e_2 W(e_1 \dot{e}_1) - \sigma_1 \Theta, \quad (3)$$

where  $\gamma > 0$ ,  $\sigma_1 > 0$ ,  $e_2 = \dot{q}_d + c_1 e_1 - x_2$ ,  $c_1$  is a positive constant vector,  $\Theta(0) = 0$ ,

$$u_r = \hat{\varepsilon} \tanh\left(\frac{e_2}{\chi}\right), \quad (4)$$

where  $u_r$  is the robust control expression that is introduced to reduce the effects of fuzzy type-2 estimation errors,

$$\dot{\hat{\varepsilon}} = \eta \hat{e}_2 \tanh\left(\frac{e_2}{\chi}\right) - \sigma_2 \hat{\varepsilon}, \quad (5)$$

where  $\eta > 0, \sigma_2 > 0, \chi > 0, \hat{e}(0) = 0$  are parameters,

$$u_p = c_2 e_2, \tag{6}$$

where  $c_2 > 0$ .

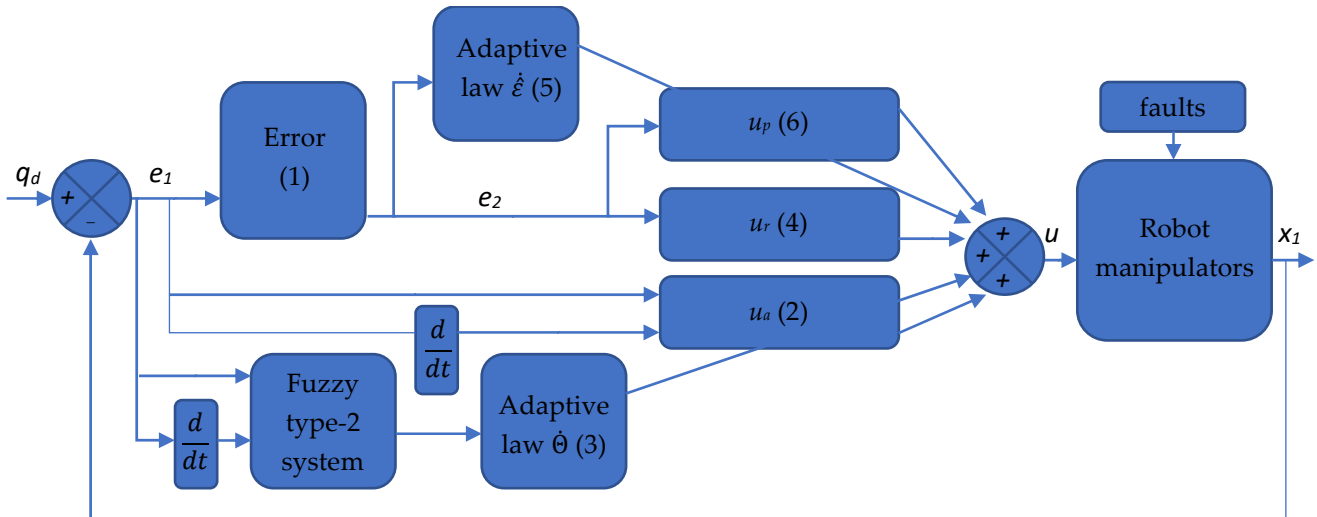


Figure 1. AFT2BC control scheme [17], where  $u$  are the desired robot axes' positions.

The authors of [20] presented a fault-tolerant control scheme in the form of non-singular terminal synergetic control (NTSC) combined with interval type-2 fuzzy satin bowerbird optimization (IT2FSBO). The paper also described an adaptive augmented extended Kalman filter (A-AEKF) that detected, identified, and isolated actuator faults, even in the presence of noise [21].

The authors of [22] presented a neuro-fuzzy (NF) robot fault-detection algorithm that allowed for the control of a robot with an SRI [23] controller despite sensor or axis actuator failures. The detection and fault-tolerance architecture of the proposed solution was built with a multilayer perceptron trained with the backpropagation of errors and a fuzzy logic block. The input layer of the artificial neural network consisted of 15 neurons organized into 3 groups, each consisting of 5 neurons, 1 for each axis. The three groups were used in turn for the preliminary assessment of the position, velocity, and acceleration. The network had two groups of output neurons, each with five neurons, one for each axis. These groups generated the positions and velocities for each axis in turn. The results of the multilayer perceptron (MLP) in the form of the product of the position and velocity, i.e.,  $q\dot{q}$ , was summed with the output parameters of the robot. This sum was sent to the inference block, along with the results of the fuzzy logic block. The inference block receiving the data provided information about the axis that was damaged. The algorithm was successfully tested on a simulation of the ER5u robot in a MATLAB environment.

The authors of [24] presented a system in which an artificial neural network in the form of a multilayer perceptron (MLP) was responsible for monitoring and fault detection (FD). This system consisted of a block in which the MLP provided information about the system failure while the fuzzy logic rule base made decisions about the type of fault and its location. The authors demonstrated the performance of fault detection based on a neuro-fuzzy (NF) application for a robotic arm with 5 degrees of freedom when one of the robot's axes fails. The architecture of the system was similar to the solution described in [23] but in this solution, the inference by the fuzzy logic system was based on the output of the neural network.

The authors of [25] proposed a new solution in the form of self-tuning fuzzy proportional-integral-derivative non-singular fast terminal sliding-mode control (SFT-PID-NFTSM) with time delay estimation (TDE). The proposed method involved the self-tuning of the

proportional–integral–derivative (PID) block of the controller using fuzzy logic algorithms. Compared to other well-known methods described in the literature, such as the PID controller or the controller with a fuzzy logic block responsible for the self-tuning of the PID block, i.e., (PID-NFTSM), the results of [25] showed that the controller proposed in this work was more stable, had less overshoot, and improved transient responses. In addition, the integration of TDE reduced the required computational power of the controller and also helped to eliminate the requirement to know the exact dynamics of the system. The architecture of this system is shown in Figure 2. In order to fully understand the operation of the presented architecture, it is necessary to explain Formulas (7)–(15), which underpin the system [25]:

$$s = e + k_1 e^{[\lambda]} + k_2 \dot{e}^{[\frac{p}{q}]}, \tag{7}$$

where  $s$  is the NFTSM sliding surface,  $e$  is the error,  $k_1 = \text{diag}(k_{11}, k_{12}, \dots, k_{1n}) \in R^{n \times n}$  and  $k_2 = \text{diag}(k_{21}, k_{22}, \dots, k_{2n}) \in R^{n \times n}$  are two positive definite matrices,  $p$  and  $q$  are two positive odd numbers selected to satisfy the conditions  $1 < \frac{p}{q} < 2$  and  $\lambda > p/q$ ,

$$S_{PID}(t) = K_p s(t) + K_i \int_0^t s(t) dt + K_d \frac{ds(t)}{dt}, \tag{8}$$

where  $S_{PID}$  is the sliding surface based on PID-NFTSM,  $K_p$ ,  $K_i$ ,  $K_d$  are the proportional, integral, and derivative gains,

$$|\Delta(x_1, x_2, u, \tau_d)| \leq \Delta_0, \tag{9}$$

where  $x_1 = q, x_2 = \dot{q}, u = \tau$  are the state variables,  $\tau_d$  is the disturbance torque,  $\Delta_0$  is a constant,  $\Delta(x_1, x_2, u, \tau_d)$  covers all the effects of the uncertainties, disturbances, and faults,

$$|\dot{\Delta}(x_1, x_2, u, \tau_d)| \leq \Delta_1, \tag{10}$$

where  $\Delta_0$  is a constant,

$$\wedge(\dot{e}, \Delta)_{(t)} \cong \wedge(\dot{e}, \Delta)_{(t-L)}, \tag{11}$$

where  $\wedge(\dot{e}, \Delta)_{(t)}$  is the unknown function,  $(t - L)$  is the time delay,

$$u_{TDE_t} \cong \widehat{\wedge}(\dot{e}, \Delta)_{(t)}, \tag{12}$$

where  $u_{TDE_t}, \widehat{\wedge}(\dot{e}, \Delta)_{(t)}$  is the estimation of the unknown function,

$$\dot{u}_{ar} = (\widehat{K} + a) \text{sign}(S_{PID}), \tag{13}$$

where  $\dot{u}_{ar}$  is the adaptive law,  $K$  is an unknown constant,  $\widehat{K}$  is the estimation of the bounded value  $K$ ,

$$u_{PID} = -\Omega^+(\dot{e}x_1)(u_{eq} + u_{TDE} + u_{ar}), \tag{14}$$

where  $u_{PID}$  is the controller output signal,  $-\Omega^+(\dot{e}x_1)$  is a lumped unknown function,

$$V = \frac{1}{2} S_{PID}^T S_{PID} + \frac{1}{2} C \check{K}^T \check{K}, \tag{15}$$

where  $V$  is the Lyapunov function,  $\check{K} = K - \widehat{K}$  is the adaption error, and  $C$  is the adaption gain.

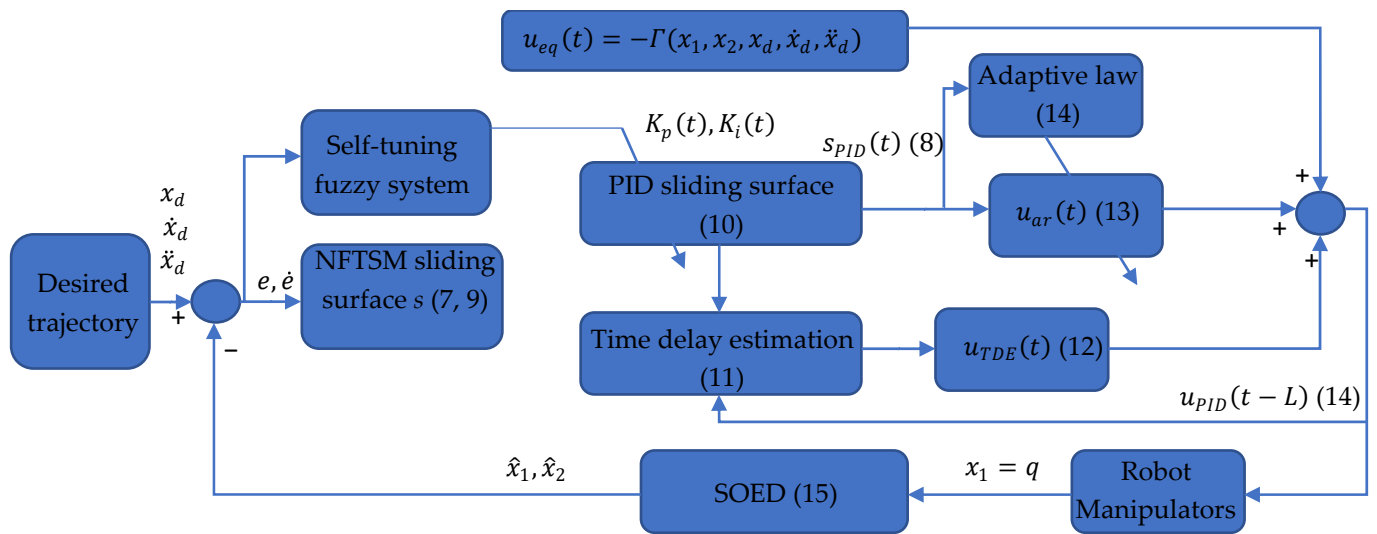


Figure 2. (SFT-PID-NFTSM) with (TDE) [25].

The authors of [26] presented a new method of fault diagnosis and fault-tolerant control for a manipulator using a combination of a support vector machine (SVM)-based neural adaptive high-order variable structure observer (ANHWSO) and an adaptive modern fuzzy backstepping variable structure controller (AMFBVSC). The architecture of this control system is shown in Figure 3. An SVM machine learning technique was used in this method for error detection and identification and the control process was based on AMFBVSC. The proposed control method, i.e., ANHVS0, improved the performance of fault identification significantly compared to the neural high-order variable, structure observer (NHVSO), and variable structure observer VSO. The work described in [26] is another example of the application of artificial intelligence methods in fault identification. In this case, neural networks and machine learning were used in control, where fuzzy logic was used to tune the proposed method.

In [27], the authors proposed the use of a dual neural network (DNN) [28] for fault diagnosis and fault tolerance. This was the first study found in the literature to present the possibility of controlling a redundant manipulator when more than two axes fail. The paper presented the realization of the task of chalking a circle when as many as three of the available seven axes fail at different time periods. Prior to this, the control task at the time of axis failure was to exclude a single axis or a single sensor.

The authors of [29] proposed and compared two approaches to FD error detection in a welding robot based on the prediction of accelerations of individual axes. The first acceleration prediction approach to fault detection was the use of a self-organizing map neural network (SOMNN). This network is shown in Figure 4a. The second approach used a radial basis function neural network (RBFNN), which is shown in Figure 4b. The paper demonstrated the feasibility of using RBFNN and SOMNN to predict the acceleration of individual axes for either detecting robot faults or predicting the remaining life of the device.

The authors of [30] presented decentralized control strategies based on a radial basis function neural network (RBFNN). In this control method, each joint was treated as a separate subsystem. A fault in the joint was detected using a velocity observer. The presented law of adaptive control was based on information from individual joints. It can be applied to any robotic arm configuration.

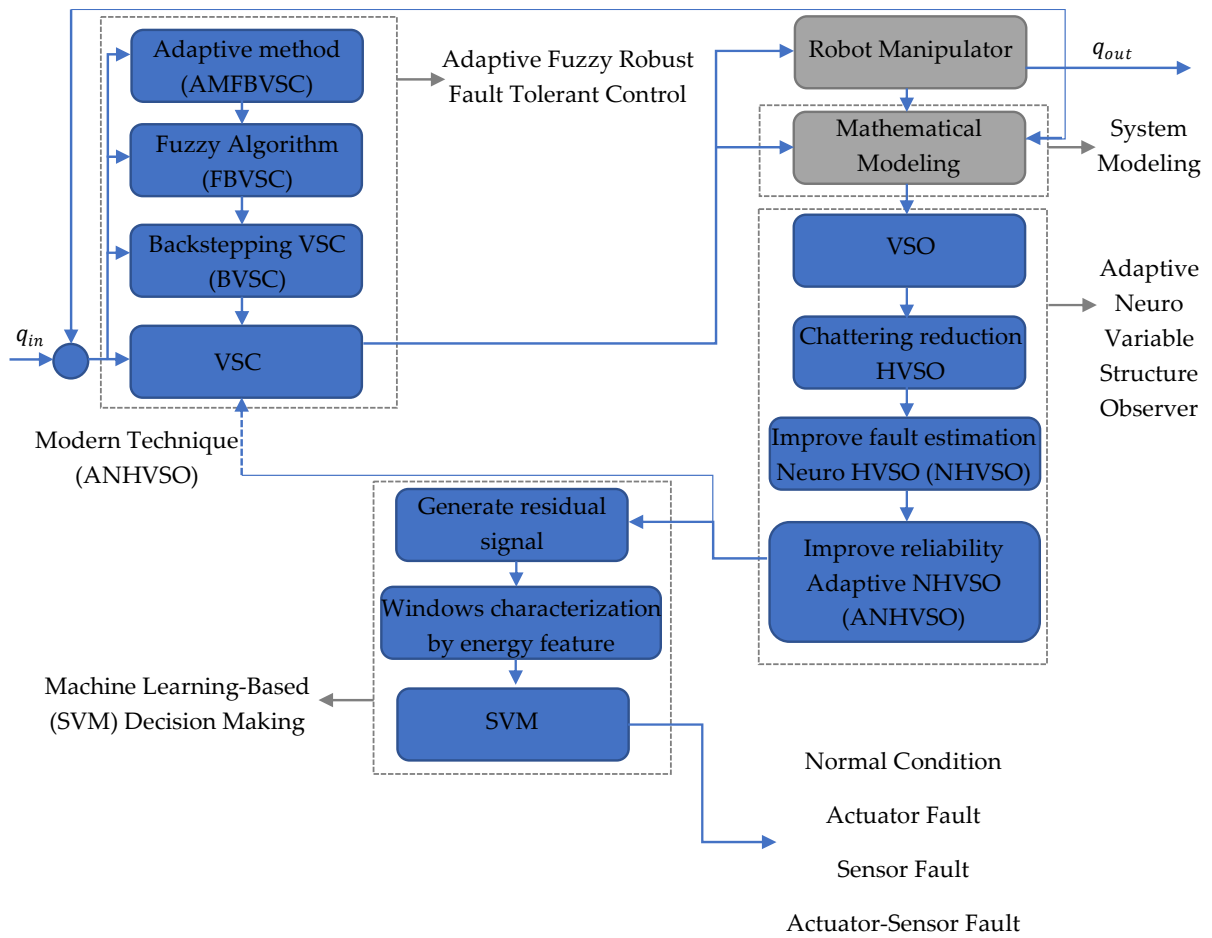


Figure 3. Control architecture with observer (ANHVS) and controller (AMFBVSC) [26].

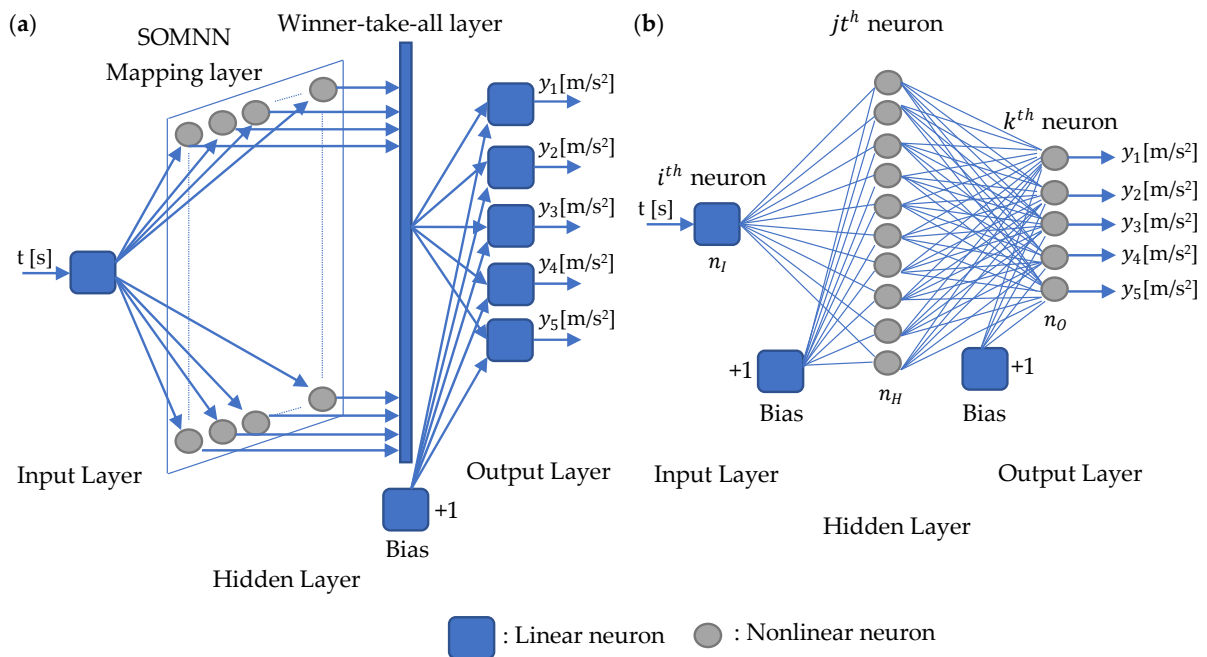


Figure 4. (a) Self-organizing map neural network (SOMNN) [29]. (b) Radial basis function neural network (RBFNN) [29].

The authors of [31] proposed a radial basis function neural network (RBFNN) for estimating system failures in order to avoid a diagnostic error. The task of the neural network was to compensate for the external disturbances and actuator failures, which were recognized in the proposed method through an adaptive disturbance observer.

In [32], an adaptive fixed-time fault-tolerant constraint control (AFTFTCC) for trajectory tracking was proposed. The system was tested by the authors on two manipulators working together. The described method differs from other methods in that it does not need components such as a state estimator or a fault observer. The method used (RBFNN) to tease out the upper limit of uncertainty.

The authors of [33] presented a control system that is robust to four types of faults: free-swinging joint faults, locked joint faults, incorrectly measured joint positions, and incorrectly measured joint velocities. Artificial neural networks in the form of MLPs and RBFNs were used to detect free-swinging defects and locked robot axes. The operation of neural networks in this solution was based on the detection of failures by the MLP, where each axis was represented by a separate MLP. The information in the form of a mapped error vector was then passed to the RBFNN, which used the vector to classify the error information. Due to the fact that the proposed control system detects the fault and then isolates it and reconfigures the controller, it was able to control the arm in case of multiple faults. However, this was not confirmed by tests in this work. The described control system is shown in Figure 5.

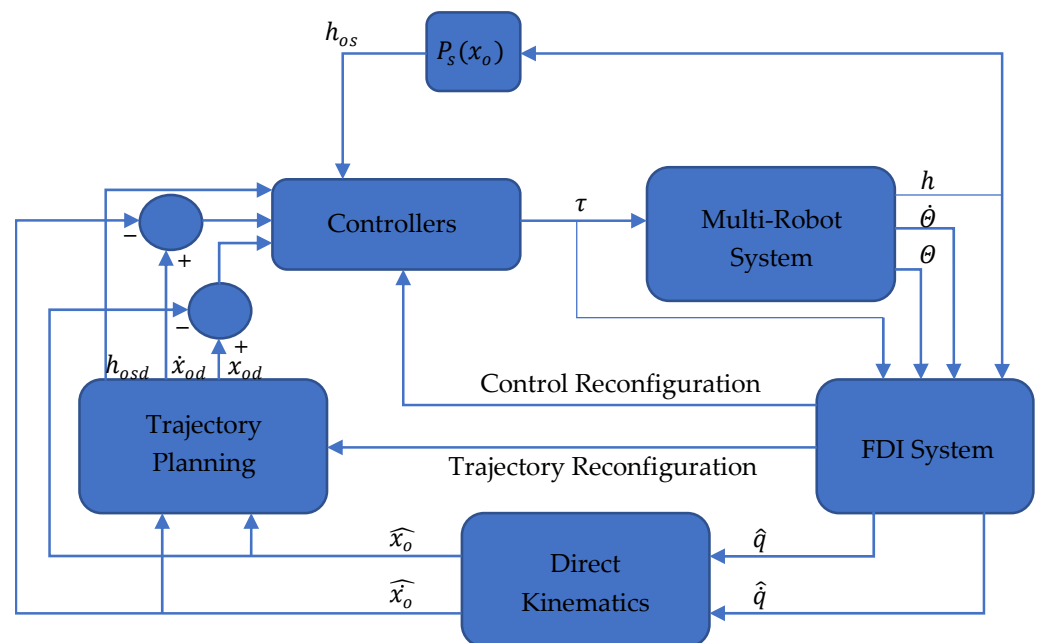


Figure 5. Robust control system architecture [33].

### 3. Sliding-Mode Control in FTCS

One of the most popular methods used to control robotic arms is sliding-mode control (SMC). Sliding-mode control is based on variable structure systems consisting of independent structures with different properties and switching logic between them. In SMC, a discontinuous control signal, for example,  $+k$  and  $-k$ , are used, which forces the system to “slide” along the cross-section of the normal characteristics of the system. When a system moves along a curved or sliding surface, the system is said to be sliding. An up-to-date overview of all the available methods of control using the sliding mode was presented in [34]. The simplest sliding-mode control method can be described by the formula:

$$u = k \cdot \text{sign}(e), \tag{16}$$

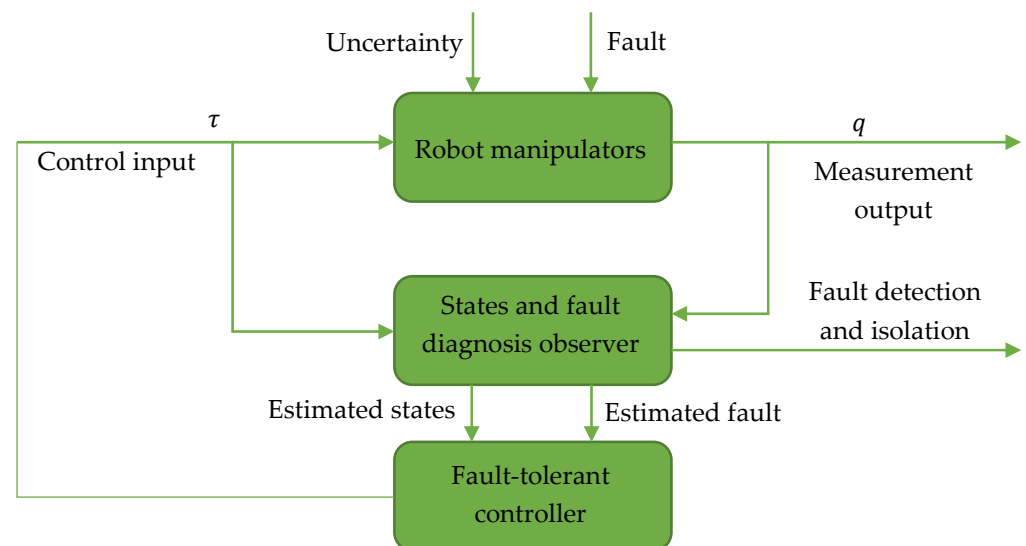


where  $u$  is the calculated signal,  $k$  is a constant that is the magnitude of change,  $e$  is the control signal error, and  $\text{sign}()$  is the sign function.

Equation (16) describes the simple sliding-mode control method. The advantage of this type of control is its simplicity and insensitivity to system uncertainty. Due to the fact that the method is discontinuous and most control systems are discrete, the main disadvantage of the presented method (16) is the system output signal oscillations called chattering. Nevertheless, the method has become attractive to researchers due to its simplicity; therefore, various sliding-mode control models are currently being designed and implemented for different objects. Due to its robustness to uncertainties, the SMC method is also often used in fault-tolerant control systems. It is applicable to manipulators for which fault tolerance is required.

One of the SMC fault-tolerant methods is the use of a sliding-mode observer (SMO) [35] for fault detection. The use of a non-linear observer makes it possible to estimate the immeasurable state and model the uncertainty, which allows the construction of an error estimation algorithm.

In [36], a fault-tolerant control system in the form of a super-twisting third-order sliding-mode (STW-TOSM) observer and a super-twisting second-order sliding mode (STW-SOSM) controller was presented. This method makes it possible to estimate errors and uncertainties without measuring the speed of the robot's members. For the correct operation of the system, only position measurements are needed. In the paper, the authors presented a comparison of the following systems: the traditional computed torque controller (CTC) [37], the active CTC-FTC controller [38], the passive SM-FTC [39] controller, and the proposed STW-SOSM-FTC controller in both passive and active forms. The test of the proposed controller showed fewer errors, better stability, and no vibrations on the obtained output waveforms. A block diagram of the arrangement of each block in the control process is presented in Figure 6.



**Figure 6.** Block diagram of the method presented by Mien Van, Pasquale Franciosa, and Dariusz Ceglarek [36].

The authors of [40] presented a system in the form of a fixed-time second-order sliding-mode observer (FxTSOSMO) and a fixed-time sliding-mode controller (FxTSMC). The work compared the proposed observer (FxTSOSMO) with the fixed-time sliding-mode observer (FxSMO) designed in an earlier work. The work compared the computed torque controller, non-singular fast terminal sliding-mode control (NFTSMC) controllers [41], and the proposed method (FxTSMC). The latter system achieved higher estimation precision

and the error of the proposed solution converged within a fixed time frame. The solution proposed in [40] is described by the following Formulas (17)–(20):

$$u_{stw} = \mu_1 |e|^{\frac{1}{2}} \text{sign}(e) + \zeta, \tag{17}$$

where  $u_{stw}$  is an approximate unknown component,  $\mu_1 > 0$ ,  $e = x_2 - v$  is a velocity error,  $\zeta = k(t)\text{sign}(e)$ ,

$$\dot{v} = \wedge u + f(x_1, x_2) + u_{stw} + k_1 [e]^{\gamma_1} + k_2 [e]^{\gamma_2}, \tag{18}$$

where  $\dot{v}$  is the observer that was designed based on a super-twisting high-order sliding-mode algorithm,  $\wedge = M^{-1}(q)$ ,  $M$  is the inertia matrix,  $k_1 > 0, k_2 > 0$ ,  $[e]^{\gamma_1} = [e]^{\gamma_1} \text{sign}(e)$ ,  $v$  is the estimate of the velocity  $x_2$ ,  $\gamma_1 = \frac{1}{2}, \gamma_2 > 2$ ,

$$s = e + \frac{1}{k_2^\gamma} \left[ \dot{e} + k_1 [e]^\alpha \right]^{\frac{1}{\gamma}}, \tag{19}$$

where  $s$  is the proposed sliding surface,  $\alpha > 1, \frac{1}{2} < \gamma < 1$ ,

$$u = u_0 + u_c + u_s, \tag{20}$$

where  $u$  is the proposed controller output,  $u_0 = \wedge^+ (-f(x_1, x_2) - u_{stw} + \ddot{x}_d - k_1 |e|^{\alpha-1} \dot{e})$ ,  $\wedge^+$  is the pseudo-inverse of the  $\wedge$ ,  $u_c = -\wedge^+ k_2^\gamma |T(e, \dot{e})|^{1-\frac{1}{\gamma}} \dot{e}$ ,  $u_s = \wedge^+ (-(\Gamma + \alpha) \text{sign}(s) - \lambda_1 [s]^{m_1} - \lambda_2 [s]^{m_2})$ ,  $\Gamma$  is a positive constant,  $0 < m_1 < 1, m_2 > 1$ .

The proposed algorithm was designed to follow a preset trajectory. The block for generating the preset trajectory is thus the first block of control. The signal from this block, together with the signal feedback from the FxTSOSMO block described by Equations (17) and (18), goes to the summing node. The feedback signal is the current positions and velocities. The position and velocity errors go from the summation node to the sliding surface block described by Equation (19). The signal  $s$  from the sliding surface block and  $u_{stw}$  from the FxTSOSMO block go to the FxTSMC block described by the equation. The output of the FxTSMC block  $u$  described by Equation (20) goes to both the FxTSOSMO block and the robot. Finally, the FxTSOSMO block retrieves information about the robot in the form of positions and velocities.

In [42,43], the authors presented a fault-tolerant control system for a single-link flexible joint manipulator (SFJM). This work proved that it is possible to control the flexible coupling in a robot joint using SMC with SMTO. The presented results show that SMC is a method that can be applied to objects characterized by non-linearity and uncertainty. However, the presented solution was validated only for a single-axis manipulator, but it is possible to develop a similar system for a manipulator with more degrees of freedom.

The authors of [44] proposed a novel methodology for manipulator adaptive backstepping non-singular fast terminal sliding-mode control (ABNFTSMC). The proposed approach combined NFTSMC with a backstepping design mechanism. The combination resulted in few tracking errors and low “chattering” on the output and provided fast response transients. The described system was compared to CTC, PID, PID-SMC, and NFTSMC controllers. The comparison of these systems showed improvements in the above-mentioned parameters.

In [45], a system consisting of combined non-singular fast terminal sliding-mode control (NFTSMC) and a high-order sliding-mode (HOSM) controller was proposed. The presented control system also used an algorithm based on time delay estimation (TDE) for fault estimation. The results described in the work showed that fast terminal sliding-mode control (FTSMC) and NFTSMC systems have faster convergence compared to the non-singular terminal sliding-mode control method. FTSMC systems may encounter the problem of singularity during operation. However, the work shows the advantage of the NFTSMC system over the above-mentioned ones. Other methods using both active

fault-tolerant control systems (AFTCS) and passive fault-tolerant control systems (PFTCS) were also rated in the paper.

In [46], a combination of a controller in the form of non-singular fast terminal sliding-mode control (NFTSMC) with an observer in the form of third-order sliding mode (TOSM) was presented. The research results showed that the TOSM observer can estimate the speed of the system so that the system does not need to measure this. This method was compared with the SMC, NTSMC, and NFTSMC methods. The proposed NFTSMC method had the fewest trajectory-tracking errors. The TOSM observer, on the other hand, proved more accurate and had less “chattering” than the SOSM observer.

The authors of [47] presented a combination of a fast terminal sliding-mode surface (FTSMS), super-twisted reaching control law (STRCL), and disturbance observer (DO). The proposed method was compared with the SMC and NFTSMC methods. The method described in the paper provided convergence in finite time and effectively combatted the chattering phenomenon. By using a disturbance observer, the complexity of the calculations was reduced. A DO was also used to estimate uncertainties in the form of dynamics, external disturbances, and failures. A diagram of the proposed method is shown in Figure 7. This work presents the new fast terminal sliding surface described by the formula in [47]:

$$\sigma_i = \dot{e}_i + \frac{2\gamma_1}{1 + E^{\mu_1(|e_i| - \Phi)}} e_i + \frac{2\gamma_2}{1 + E^{\mu_2(|e_i| - \Phi)}} |e_i|^\alpha \text{sign}(e_i), \tag{21}$$

where  $\sigma \in R$  is the FTSMS,  $E$  is an exponential function,  $e_i$  represents the positional control error,  $\dot{e}_i$  represents the velocity control error,  $\gamma_1, \gamma_2, \mu_1, \mu_2$  are the positive constants,  $0 < \alpha < 1, \Phi = \left(\frac{\gamma_1}{\gamma_2}\right)^{1/(1-\alpha)}$ ,

$$u_r = Y_1 |\sigma|^{0.5} \text{sign}(\sigma) + \eta, \tag{22}$$

where  $u_r$  is STRCL,  $Y_1 = \text{diag}(Y_{11}, \dots, Y_{1n})$ .

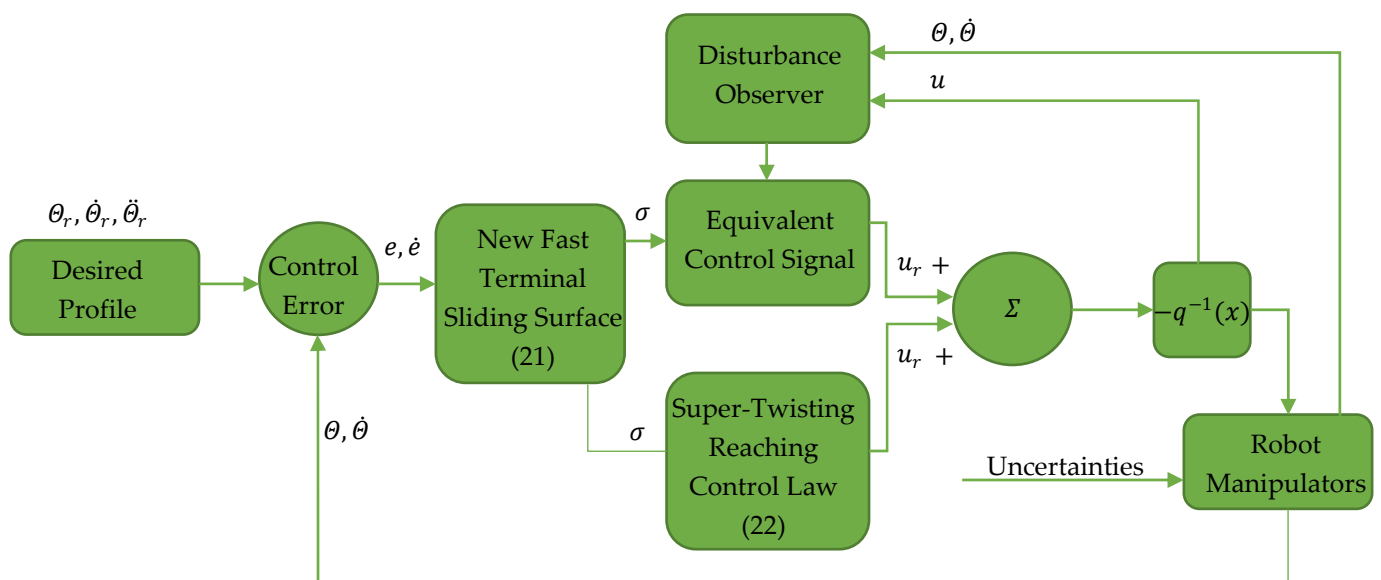


Figure 7. Diagram FTSMs-STRCL-DO [47].

The authors of [48] proposed a new sliding-mode control technique, namely active fault-tolerant control with synchronous sliding-mode control AFTC-SSMC. This control relied on the fact that with a traditional SMC controller, only the position error converges to zero. In synchronous control, on the other hand, both the position error and kinematic relationship between the errors converge to zero. The method was compared with standard active and passive sliding-mode control, namely AFTC-SMC and PFTC-SMC.

It was shown that the pro-rated method had better fault tolerance and provided better trajectory-tracking performance.

The authors of [49] presented a fault-tolerant manipulator control strategy in the form of a combination of NFTSMC and the proposed high-speed third-order sliding-mode observer TOSMO. This combination was able to handle unknown input data, further reducing the vibration phenomenon and improving the accuracy of trajectory tracking. The proposed high-speed TOSMO could estimate both the velocity signal and input data faster than a standard TOSMO. The structure of such a system is similar to that of the FTSMC-STRCL-DO [47] but it has a different observer model. The structure of the proposed method is shown in Figure 8. The solution proposed by the authors is described by the following formulas [49]:

$$\hat{s} = \hat{e} + \int [\beta_1 |e|^{\gamma_1} \text{sign}(e) + \beta_2 \hat{e}^{\gamma_2} \text{sign}(\hat{e}) + \beta_3 e + \beta_4 e^3] dt, \tag{23}$$

where  $\hat{s}$  is the NFTS surface,  $\beta_1, \beta_2, \beta_3, \beta_4$  are positive constants,  $0 < \gamma_1 < 1, \gamma_2 = 2\gamma_1 / (1 + \gamma_1)$  are constants,

$$u_{eq} = \psi(x) + k_2 |\tilde{x}_1|^{\frac{1}{3}} \text{sign}(\tilde{x}_1) + \Gamma(\dot{\tilde{x}}_1 - x_2) + \int k_3 \text{sign}(\tilde{x}_1) + A - \ddot{x}_d, \tag{24}$$

where  $u_{eq}$  is the equivalent control law,  $A = \beta_1 |e|^{\gamma_1} \text{sign}(e) + \beta_2 \hat{e}^{\gamma_2} \text{sign}(\hat{e}) + \beta_3 e + \beta_4 e^3$ ,  $\psi(x) = M(q)^{-1} [-C(q, \dot{q}) - G(q)]$  is the nominal model of the robot manipulator,  $\Gamma$  is a positive constant,  $k_2, k_3$  denotes the sliding gains,  $\ddot{x}_d$  is an expected acceleration

$$u_{sw} = (\Delta_\delta + \mu) \text{sign}(\hat{s}), \tag{25}$$

where  $u_{sw}$  is the switching control law,  $\mu$  is a small positive constant, and  $\Delta_\delta$  is the estimation error as a positive value.

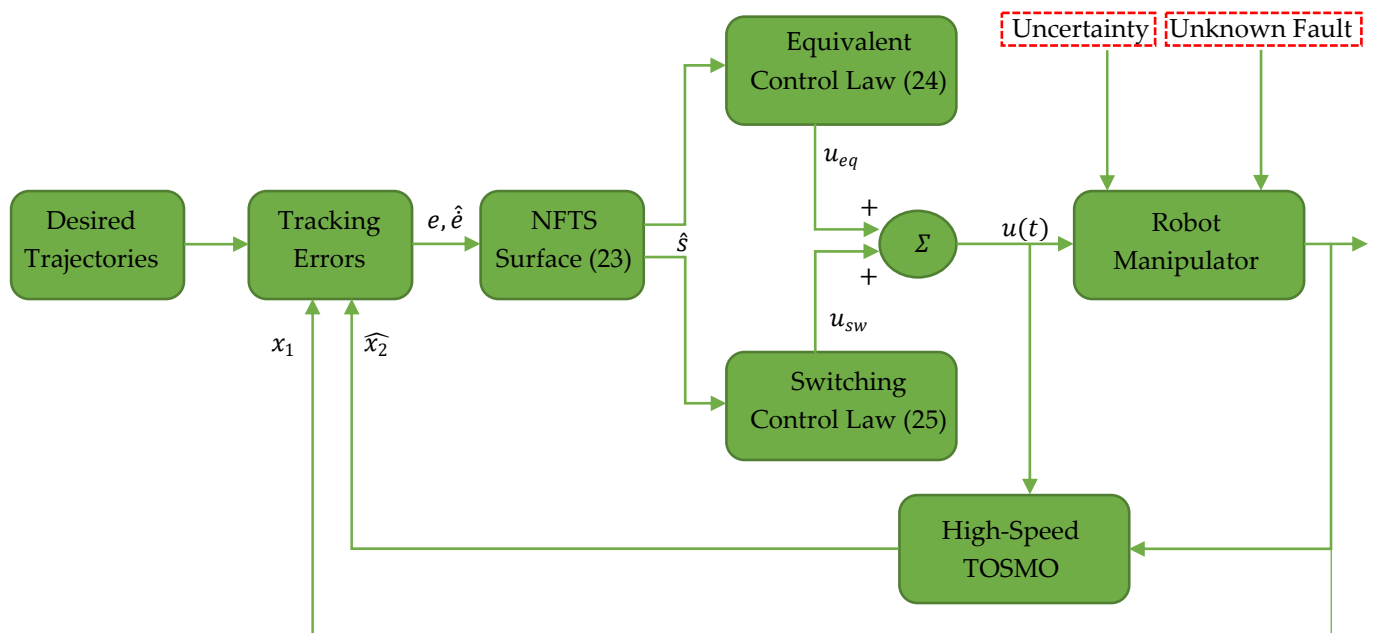


Figure 8. Structure of the NFTSMC-TOSMO control method [49].

#### 4. Other Robot-Dedicated FTCS

In addition to the leading fault-tolerant manipulator control methods such as SMC and artificial intelligence, other dedicated manipulator control methods can also be found in the literature. Some of them are presented in this section of the article.

The authors of [50] presented the use of the well-known Newton–Raphson Method (NRM) [51] to control a manipulator when a robot axis fails. The authors of this paper attached a QR code to the robot’s end arm. This code was read by a camera located at the base of the robot and then the position of the robot’s end was fed into the control software. This allowed the algorithm to determine the moment at which the failure occurred by not matching the position tracked by the camera with the calculated position. Then, the system, which used the Newton–Raphson method in the inverse kinematics calculation to find a good approximation for the root of a real-valued function, determined which joint had failed. To determine the faulty axis, the position measured by the camera was compared with the position calculated by the NRM. If a difference was detected, the system checked which axis was damaged. If a faulty joint was detected, the failed joint was treated as rigid and the robot’s inverse kinematics were recalculated using the Newton–Raphson method. Using this solution, the authors of the paper proved that if the robot was able to reach the set position despite the loss of an axis, the algorithm was able to help the robot achieve its goal.

In [52], a new method of hybrid fault-tolerant predictive control (HFTPC) was proposed. This method was tested on a hybrid manipulator that was a combination of hydraulic and mechanical elements. The paper compared the proposed HFTPC method, one of the components of which was a predictive control (MPC) model [53], with control in which only the MPC part was running. The presented system was able to control a hybrid manipulator when a failure occurred.

The authors of [54] presented robust fault-tolerant tracking control (RFTTC) schemes for uncertain non-linear feedback systems (NFS) using operator theory-based robust right coprime factorization (ORRCF). It was shown that both RFTTC based on an internal model and RFTTC based on operator compensation were effective in dealing with interference and erroneous signals. The proposed system was tested on a trajectory-tracking task by a two-axis robot.

The authors of [55] presented an adaptive fault-tolerant control system for unknown actuator faults. Detailed fault information was not required for the algorithm to work. The proposed method omitted complex logarithmic transformations in the pre-scribed performance control (PPC) structure [56], making the controller efficient and simple to implement.

The authors of [57] proposed a robust fixed-time fault-tolerant control system (RFTFTC) by introducing an additional auxiliary vector that was described by Formula (26) [57]:

$$\chi = \dot{e} + \lambda e, \tag{26}$$

where  $\lambda \in R^+$ ,  $e = q - q_d$  is the position error and  $\dot{e} = \dot{q} - \dot{q}_d$  is the velocity error.

A schematic of the proposed system is shown in Figure 9.

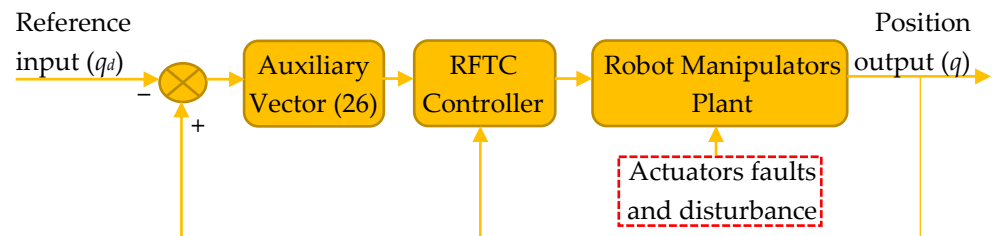


Figure 9. RFTC with auxiliary vector [57].

The authors of [58,59] presented an FTCS for manipulators based on an active inference controller (AIC). AIC makes use of the sensory prediction error in the free energy for the generation of residuals and thresholds for FDI. It does not require additional controllers for fault recovery. In [59], the AIC was improved, and unbiased AIC (u-AIC) was introduced to reduce the probability of false positives and allow for the easy definition of probabilistically robust thresholds for fault detection.

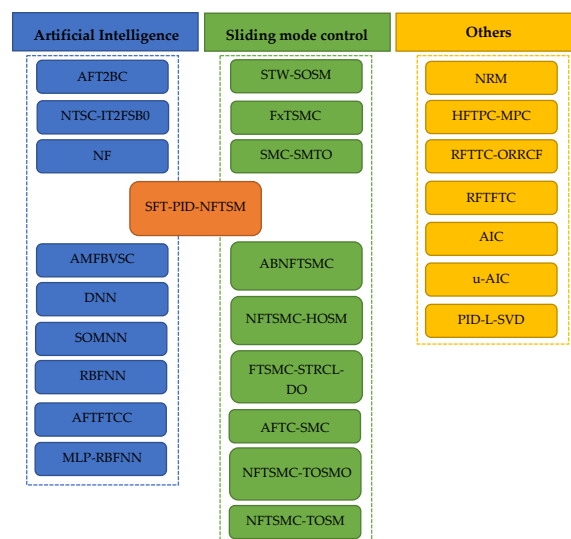
The authors of [43] presented fault-tolerant control using a proportional–integral–derivative (PID) controller. In addition to the PID controller, the system consisted of a Luenberger observer [60], which was able to estimate the errors of the FDI analysis. From the information provided by the FDI, the magnitude of the error was evaluated using singular value decomposition (SVD) [61]. The system in this solution was tested on a single-link flexible manipulator.

In earlier works, i.e., before 2019, the algorithms described in the literature for the control of robots in the case of communication faults dealt with two control types, i.e., state observers or estimators. The works in [62,63] presented a completely different approach to fault-tolerant control. The authors of these works dealt with a problem involving interference in the communication path through which information is sent between the robot’s actuators and the controller. The paper proposed to re-construct data corrupted during transmission using a splicing code. The transmission was carried out using the Controller Area Network (CAN) protocol [64].

The authors of [65] presented as many as four techniques for dealing with controlled oscillation and axis damage. These techniques were sinh-cosh, neural compensation, PID gain scheduling, and sinh-cosh gain scheduling. All of them were tested on a manipulator in which the failure was detected by measuring the currents through the robot axis drives. At the moment of axis failure, the current dropped to zero and then the axis was treated as passive, i.e., not participating in control. In the results of the paper, all four methods were compared. The comparison of the methods was conducted using a presentation of the trajectory-tracking error for each method. The best results in terms of the position errors of the individual axes were obtained for neural compensation, whereas the worst results were obtained by the sinh-cosh gain-scheduling method.

### 5. Summary

This paper presented an overview of the investigated FTC-type solutions used in manipulators. The works that were considered and described were mainly from the last five years. The division of the literature items presented in this article, as proposed by the authors, is shown in Figure 10. The leading solutions are AI (artificial intelligence) and SMC (sliding mode control). A fault-tolerant control system that is a combination of AI and SMC has also been developed, namely SFT-PID-NFTSM [25]. Other presented methods are based on PID controllers [53], NRM [50], MPC [52], and AIC [58,59], or are proposals that deal with interference occurring in the CAN communication protocol [62,63].



**Figure 10.** Proposed breakdown of fault-tolerant control system methods used in manipulators. All abbreviations of the control methods and their references to the literature are in the text above and in Table 1 below.

**Table 1.** Comparison of FTC methods used for robot control described in the cited articles.

	Accuracy	Robustness	Dependence on Model or Training Data	Implementation Complexity	Computational Cost
AFT2BC [17]	2	4	2	3	2
NTSC-IT2FSB0 [20]	5	4	3	3	3
NF [22,24]	2	3	5	4	5
AMFBVSC [26]	5	3	5	5	5
DNN [27]	5	5	4	4	4
SOMNN [29]	3	3	5	4	4
RBFNN [29–31]	5	4	3	4	4
AFTFTCC [32]	4	4	3	4	4
MLP-RBFNN [33]	3	4	5	4	5
SFT-PID-NFTSM [25]	2	4	2	3	3
STW-SOSM [36]	3	4	1	2	2
FXTSMC [40]	4	5	1	2	2
SMC-SMTO [42]	-	-	1	2	2
ABNFTSMC [44]	3	5	1	3	2
NFTSMC-HOSM [45]	3	4	1	2	2
FTSMC-STRCL-DO [47]	4	5	1	2	1
AFTC-SMC [48]	5	5	1	2	2
NFTSMC-TOSMO [49]	4	5	1	2	2
NFTSMC-TOSM [46]	4	5	1	2	2
NRM [50]	2	3	2	3	4
HFTPC-MPC [52]	2	3	3	3	3
RFTTC-ORRCF [54]	5	5	4	4	3
RFTFTC [57]	4	4	2	2	2
AIC [58]	4	4	3	3	3
U-AIC [59]	4	5	3	3	3
PID-L-SVD [43]	-	-	-	2	2

**Accuracy:** Rating scale (1–5), where 1 indicates low accuracy and 5 indicates high accuracy. **Robustness:** Rating scale (1–5), where 1 indicates low robustness and 5 indicates high robustness. **Dependence on model or training data:** Rating scale (1–5), where 1 indicates no dependence on training data and 5 indicates high dependence on training data. **Implementation complexity:** Rating scale (1–5), where 1 indicates that the implementation was simple and 5 indicates that the implementation was very complex. **Computational cost:** Rating scale (1–5), where 1 indicates that the cost was small and 5 indicates that the cost was high.

The purpose of this review was to present the issue of FTC in the context of manipulators, particularly for the control of a robot in the case of a one- or two-axis failure. We decided to collect works from the last 20 years and then group them. In this way, the three groups created related to the sections of this paper, i.e., AI-based methods, SMC-based methods, and other methods, which, numerically, constituted a similar number to the first two groups. The goal of our review was to systematically collect and summarize the existing research on the topic of FTC in manipulators. After reading this paper, it is expected that readers will have gained a general understanding of FTC in manipulators. We believe that the topic of FTC is very subjective and depends strictly on the application, e.g., if a robot is connected to very long wires, it may be that using FTC algorithms for the communication line is a much better solution than using an AI or SMC-based controller. For manipulators with a large number of degrees of freedom, a computationally simple SMC-based controller is probably better but this may also depend on the required motion speeds or positioning accuracy. The purpose of this article is to direct the reader to the solutions and the valuable works of the cited authors. Therefore, the methods described in the cited articles are compared in Table 1. Each of the proposed methods is valid for a different type of application and their comparison could only be performed for detailed boundary conditions.

## 6. Conclusions

The study presented in this paper deals with FTCS in relation to manipulators. The authors of the papers discussed in this review deal with failures in both single and multiple axes. In addition, they address aspects of tolerating erroneous position sensor readings and propose algorithms to deal with interference occurring on the communication line between the controller and robot actuators. A review of the literature shows that the topic of FTCS as applied to manipulators is currently being strongly developed. The use of FTCS systems allows the optimization of production by avoiding downtime caused by individual component failures in robotic arms. Looking toward the future, several opportunities for the application of FTC in manipulators have been identified by the authors. The first

possibility is for the robot to react to obstacles in its working environment. However, no studies have yet been presented in which the FTC algorithm was implemented and the robot was able to avoid obstacles. Another possible future use of FTC algorithms is the cooperation between a robot and a human. However, there are no studies that indicate how the FTC algorithm would behave in such a scenario. Additionally, there are currently no studies that demonstrate how FTC algorithms could facilitate cooperation between robots such as delegating work to a more efficient robot when another robot is defective or robots communicating with each other about the detection of defects.

**Author Contributions:** Conceptualization, P.N. and A.M.; methodology, P.N. and A.M.; resources, P.N.; data curation, P.N.; writing—original draft preparation, A.M. and P.N.; writing—review and editing, A.M. and P.N.; visualization, P.N.; supervision, P.N. and A.M.; joint first authorship, A.M.; project administration, A.M.; funding acquisition, A.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Polish Ministry of Science and Education grant no. 0614/SBAD/1565.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Aviziens, A. Fault-tolerant systems. *IEEE Trans. Comput.* **1976**, *100*, 1304–1312. [[CrossRef](#)]
2. Stengel, R.F. Intelligent failure-tolerant control. *IEEE Control. Syst. Mag.* **1991**, *11*, 14–23.
3. Veillette, R.J.; Medanic, J.V.; Perkins, W.R. Design of reliable control systems. In Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, USA, 5–7 December 1990; pp. 1131–1136.
4. Patton, R. Robustness issues in fault-tolerant control. In Proceedings of the IEE Colloquium on Fault Diagnosis and Control System Reconfiguration, London, UK, 26 May 1993; p. 1.
5. Patton, R.J. Fault-tolerant control: The 1997 situation. *IFAC Proc. Vol.* **1997**, *30*, 1029–1051. [[CrossRef](#)]
6. Maciejewski, A.A. Fault tolerant properties of kinematically redundant manipulators. In Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13–18 May 1990; pp. 638–642.
7. Chladek, J.T. Fault tolerance for space based manipulator mechanisms and control system. In Proceedings of the First International Symposium On Measurement and Control in Robotics, Houston, TX, USA, 20–22 June 1990.
8. Tesar, D.; Sreevijayan, D.; Price, C. Four-level fault tolerance in manipulator design for space operations. In Proceedings of the First International Symposium on Measurement and Control in Robotics, Houston, TX, USA, 20–22 June 1990.
9. Wu, E.; Diftler, M.; Hwang, J.; Chladek, J. A fault tolerant joint drive system for the space shuttle remote manipulator system. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; pp. 2504–2509.
10. Tesar, D. On the design of fault-tolerant robotic manipulator systems. In Proceedings of the 6th Annual Workshop on Space Operations Applications and Research (SOAR 1992), Victorville, CA, USA, 4–6 August 1992; Volume 2, p. 223.
11. Visinsky, M.L.; Cavallaro, J.R.; Walker, I.D. *Robotic Fault Tolerance: Algorithms and Architectures*; Prentice Hall: Englewood Cliffs, NJ, USA, 1993.
12. Ting, Y.; Tosunoglu, S.; Tesar, D. A control structure for fault-tolerant operation of robotic manipulators. In Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 2–6 May 1993; pp. 684–690.
13. Selkainaho, J.; Halme, A. An intelligent fault tolerant control system. *IFAC Proc. Vol.* **1987**, *20*, 69–73. [[CrossRef](#)]
14. Hörmann, A.; Hugel, T.; Meier, W. A concept for an intelligent and fault-tolerant robot system. *J. Intell. Robot. Syst.* **1988**, *1*, 259–286. [[CrossRef](#)]
15. Groom, K.N.; Maciejewski, A.A.; Balakrishnan, V. Real-time failure-tolerant control of kinematically redundant manipulators. *IEEE Trans. Robot. Autom.* **1999**, *15*, 1109–1115. [[CrossRef](#)]
16. Shin, J.H.; Lee, J.J. Fault detection and robust fault recovery control for robot manipulators with actuator failures. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C), Detroit, MI, USA, 10–15 May 1999; Volume 2, pp. 861–866.
17. Rahali, H.; Zeghlache, S.; Benyettou, L. Fault Tolerant Control of Robot Manipulators Based on Adaptive Fuzzy Type-2 Backstepping in Attendance of Payload Variation. *Int. J. Intell. Eng. Syst.* **2021**, *14*, 312–325. [[CrossRef](#)]
18. Ali, K.; Mehmood, A.; Iqbal, J. Fault-tolerant scheme for robotic manipulator—Nonlinear robust back-stepping control with friction compensation. *PLoS ONE* **2021**, *16*, e0256491. [[CrossRef](#)]



19. Tanaka, K.; Hori, T.; Wang, H.O. A multiple Lyapunov function approach to stabilization of fuzzy control systems. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 582–589. [[CrossRef](#)]
20. Hagh, Y.S.; Asl, R.M.; Fekih, A.; Wu, H.; Handroos, H. Active fault-tolerant control design for actuator fault mitigation in robotic manipulators. *IEEE Access* **2021**, *9*, 47912–47929. [[CrossRef](#)]
21. Asl, R.M.; Hagh, Y.S.; Handroos, H. Adaptive extended Kalman filter designing based on non-singular fast terminal sliding mode control for robotic manipulators. In Proceedings of the 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 6–9 August 2017; pp. 1670–1675.
22. Dev Anand, M.; Selvaraj, T.; Kumanan, S.; Ajitha, T. A neuro-fuzzy-based fault detection and fault tolerance methods for industrial robotic manipulators. *Int. J. Adapt. Innov. Syst.* **2010**, *1*, 334–371. [[CrossRef](#)]
23. Deo, A.S. Application of Optimal Damped Least-Squares Method to Inverse Kinematics of Robotic Manipulators. Master's Thesis, Rice University, Houston, TX, USA, 1991.
24. Dev Anand, M.; Selvaraj, T.; Kumanan, S. Fault Detection and Fault Tolerance Methods for Industrial Robot Manipulators Based on Hybrid Intelligent Approach. *Adv. Prod. Eng. Manag.* **2012**, *7*, 225–236. [[CrossRef](#)]
25. Van, M.; Do, X.P.; Mavrouniotis, M. Self-tuning fuzzy PID-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. *ISA Trans.* **2020**, *96*, 60–68. [[CrossRef](#)] [[PubMed](#)]
26. Piltan, F.; Prosvirin, A.E.; Sohaib, M.; Saldivar, B.; Kim, J.M. An SVM-based neural adaptive variable structure observer for fault diagnosis and fault-tolerant control of a robot manipulator. *Appl. Sci.* **2020**, *10*, 1344. [[CrossRef](#)]
27. Li, Z.; Li, C.; Li, S.; Cao, X. A fault-tolerant method for motion planning of industrial redundant manipulator. *IEEE Trans. Ind. Inform.* **2019**, *16*, 7469–7478. [[CrossRef](#)]
28. Zhang, Y.; Wang, J.; Xia, Y. A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. *IEEE Trans. Neural Netw.* **2003**, *14*, 658–667. [[CrossRef](#)] [[PubMed](#)]
29. Eski, I.; Erkaya, S.; Savas, S.; Yildirim, S. Fault detection on robot manipulators using artificial neural networks. *Robot. Comput.-Integr. Manuf.* **2011**, *27*, 115–123. [[CrossRef](#)]
30. Zhao, B.; Li, Y. Local joint information based active fault tolerant control for reconfigurable manipulator. *Nonlinear Dyn.* **2014**, *77*, 859–876. [[CrossRef](#)]
31. Zhang, S.; Yang, P.; Kong, L.; Chen, W.; Fu, Q.; Peng, K. Neural networks-based fault tolerant control of a robot via fast terminal sliding mode. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 4091–4101. [[CrossRef](#)]
32. Hwang, C.L.; Yu, W.S. Tracking and cooperative designs of robot manipulators using adaptive fixed-time fault-tolerant constraint control. *IEEE Access* **2020**, *8*, 56415–56428. [[CrossRef](#)]
33. Tinós, R.; Terra, M.H.; Bergerman, M. A fault tolerance framework for cooperative robotic manipulators. *Control. Eng. Pract.* **2007**, *15*, 615–625. [[CrossRef](#)]
34. Yu, X.; Feng, Y.; Man, Z. Terminal sliding mode control—An overview. *IEEE Open J. Ind. Electron. Soc.* **2020**, *2*, 36–52. [[CrossRef](#)]
35. Khireddine, M.S.; Boutarfa, A. Fault tolerant control on robotic manipulator using sliding mode observers. In Proceedings of the 2013 International Conference on Computer Applications Technology (ICCAT), Sousse, Tunisia, 20–22 January 2013; pp. 1–6.
36. Van, M.; Franciosa, P.; Ceglarek, D. Fault diagnosis and fault-tolerant control of uncertain robot manipulators using high-order sliding mode. *Math. Probl. Eng.* **2016**, *2016*, 7926280. [[CrossRef](#)]
37. Piltan, F.; Mirzaei, M.; Shahriari, F.; Nazari, I.; Emamzadeh, S. Design baseline computed torque controller. *Int. J. Eng.* **2012**, *6*, 129–141.
38. Tosunoglu, S. Fault-tolerant control of mechanical systems. In Proceedings of the IECON'95-21st Annual Conference on IEEE Industrial Electronics, Orlando, FL, USA, 6–10 November 1995; Volume 1, pp. 127–132.
39. Merheb, A.R.; Noura, H.; Bateman, F. Design of passive fault-tolerant controllers of a quadrotor based on sliding mode theory. *Int. J. Appl. Math. Comput. Sci.* **2015**, *25*, 561–576. [[CrossRef](#)]
40. Van, M.; Ceglarek, D. Robust fault tolerant control of robot manipulators with global fixed-time convergence. *J. Frankl. Inst.* **2021**, *358*, 699–722. [[CrossRef](#)]
41. Xu, B.; Zhang, L.; Ji, W. Improved non-singular fast terminal sliding mode control with disturbance observer for PMSM drives. *IEEE Trans. Transp. Electrif.* **2021**, *7*, 2753–2762. [[CrossRef](#)]
42. Chen, Y.; Guo, B. Sliding mode fault tolerant tracking control for a single-link flexible joint manipulator system. *IEEE Access* **2019**, *7*, 83046–83057. [[CrossRef](#)]
43. Abd Latip, S.F.; Husain, A.R.; Ahmad, M.N.; Mohamed, Z. Fault tolerant control for sensor fault of a single-link flexible manipulator system. *J. Teknol.* **2016**, *78*. [[CrossRef](#)]
44. Van, M.; Mavrouniotis, M.; Ge, S.S. An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1448–1458. [[CrossRef](#)]
45. Van, M.; Ge, S.S.; Ren, H. Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control. *IEEE Trans. Cybern.* **2016**, *47*, 1681–1693. [[CrossRef](#)] [[PubMed](#)]
46. 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](#)]
47. Vo, A.T.; Kang, H.J. A novel fault-tolerant control method for robot manipulators based on non-singular fast terminal sliding mode control and disturbance observer. *IEEE Access* **2020**, *8*, 109388–109400. [[CrossRef](#)]

48. 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](#)]
49. Nguyen, V.C.; Tran, X.T.; Kang, H.J. A Novel High-Speed Third-Order Sliding Mode Observer for Fault-Tolerant Control Problem of Robot Manipulators. *Actuators* **2022**, *11*, 259. [[CrossRef](#)]
50. Zhang, Y.; Zhu, W.; Rosendo, A. QR code-based self-calibration for a fault-tolerant industrial robot arm. *IEEE Access* **2019**, *7*, 73349–73356. [[CrossRef](#)]
51. Ge, D. Kinematics modeling of redundant manipulator based on screw theory and Newton-Raphson method. *J. Phys. Conf. Ser.* **2022**, *2246*, 012068. [[CrossRef](#)]
52. Zahaf, A.; Bououden, S.; Chadli, M.; Chemachema, M. Robust fault tolerant optimal predictive control of hybrid actuators with time-varying delay for industrial robot arm. *Asian J. Control.* **2022**, *24*, 1–15. [[CrossRef](#)]
53. Altan, A.; Hacıoğlu, R. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. *Mech. Syst. Signal Process.* **2020**, *138*, 106548. [[CrossRef](#)]
54. Bu, N.; Pang, J.; Deng, M. Robust fault tolerant tracking control for the multi-joint manipulator based on operator theory. *J. Frankl. Inst.* **2020**, *357*, 2696–2714. [[CrossRef](#)]
55. 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](#)]
56. Ma, H.; Zhou, Q.; Li, H.; Lu, R. Adaptive prescribed performance control of a flexible-joint robotic manipulator with dynamic uncertainties. *IEEE Trans. Cybern.* **2021**, *52*, 12905–12915. [[CrossRef](#)] [[PubMed](#)]
57. Liu, L.; Zhang, L.; Wang, Y.; Hou, Y. A novel robust fixed-time fault-tolerant tracking control of uncertain robot manipulators. *IET Control. Theory Appl.* **2021**, *15*, 195–208. [[CrossRef](#)]
58. Pezzato, C.; Baioumy, M.; Corbato, C.H.; Hawes, N.; Wisse, M.; Ferrari, R. Active inference for fault tolerant control of robot manipulators with sensory faults. In Proceedings of the International Workshop on Active Inference, Ghent, Belgium, 14 September 2020; Springer: Cham, Switzerland, 2020; pp. 20–27.
59. Baioumy, M.; Pezzato, C.; Ferrari, R.; Corbato, C.H.; Hawes, N. Fault-tolerant control of robot manipulators with sensory faults using unbiased active inference. In Proceedings of the 2021 European Control Conference (ECC), Delft, The Netherlands, 29 June–2 July 2021; pp. 1119–1125.
60. Brivadis, L.; Andrieu, V.; Serres, U. Luenberger observers for discrete-time nonlinear systems. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 3435–3440.
61. Epps, B.P.; Krivitzky, E.M. Singular value decomposition of noisy data: Noise filtering. *Exp. Fluids* **2019**, *60*, 126. [[CrossRef](#)]
62. Urrea, C.; Kern, J.; López-Escobar, R. Design and implementation of a fault-tolerant system for industrial robots under hostile operating conditions. *Comput. Electr. Eng.* **2021**, *90*, 106951. [[CrossRef](#)]
63. Urrea, C.; Kern, J.; López, R. Fault-tolerant communication system based on convolutional code for the control of manipulator robots. *Control. Eng. Pract.* **2020**, *101*, 104508. [[CrossRef](#)]
64. Lokman, S.F.; Othman, A.T.; Abu-Bakar, M.H. Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 184. [[CrossRef](#)]
65. Urrea, C.; Kern, J.; Álvarez, E. Design and implementation of fault-tolerant control strategies for a real underactuated manipulator robot. *Complex Intell. Syst.* **2022**, *8*, 5101–5123. [[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.