



# Article Fault-Tolerant Control of Autonomous Underwater Vehicle Actuators Based on Takagi and Sugeno Fuzzy and Pseudo-Inverse Quadratic Programming under Constraints

Zimu Zhang<sup>1</sup>, Yunkai Wu<sup>2,\*</sup>, Yang Zhou<sup>3</sup> and Dahai Hu<sup>2</sup>

- School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China; 3210507016@stmail.ujs.edu.cn
- <sup>2</sup> College of Automation, Jiangsu University of Science and Technology, Zhenjiang 212100, China; hu1350118711@163.com
- <sup>3</sup> School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212100, China; jsjxy\_zy@just.edu.cn
- \* Correspondence: wuyunkaischolar@just.edu.cn

Abstract: Autonomous Underwater Vehicles (AUVs) play a significant role in ocean-related research fields as tools for human exploration and the development of marine resources. However, the uncertainty of the underwater environment and the complexity of underwater motion pose significant challenges to the fault-tolerant control of AUV actuators. This paper presents a fault-tolerant control strategy for AUV actuators based on Takagi and Sugeno (T-S) fuzzy logic and pseudo-inverse quadratic programming under control constraints, aimed at addressing potential actuator faults. Firstly, considering the steady-state performance and dynamic performance of the control system, a T-S fuzzy controller is designed. Next, based on the redundant configuration of the actuators, the propulsion system is normalized, and the fault-tolerant control of AUV actuators is achieved using the pseudo-inverse method under thrust allocation. When control is constrained, a quadratic programming approach is used to compensate for the input control quantity. Finally, the effectiveness of the fuzzy control and fault-tolerant control allocation methods studied in this paper is validated through mathematical simulation. The experimental results indicate that in various fault scenarios, the pseudo-inverse combined with a nonlinear quadratic programming algorithm can compensate for the missing control inputs due to control constraints, ensuring the normal thrust of AUV actuators and achieving the expected fault-tolerant effect.

Keywords: AUV actuators; fault-tolerant control; T-S fuzzy; quadratic programming; control constraints

# 1. Introduction

Autonomous Underwater Vehicles (AUVs) play a crucial role in the exploration and utilization of marine resources and space. As a new tool for exploring the ocean, AUVs significantly enhance the efficiency of underwater detection and operations. They play an extensive role in various areas, such as marine resource exploration, underwater engineering operations, scientific research and surveys. As AUVs need to work long-term in the ocean, the harsh conditions and complex and ever-changing environment may cause various faults in the actuators of their motion control systems. Therefore, fault-tolerant control technology has become an urgent need to enhance the safety and reliability of AUVs.

Fault tolerance refers to a system's ability to detect and diagnose its own faults in the event of component faults, such as actuators, sensors or controlled objects, and to initiate appropriate corrective actions based on the diagnosis results. This enables the system to continue functioning as intended while maintaining performance consistency with a fault-free scenario. Fault-tolerant control can be categorized into passive fault-tolerant control and active fault-tolerant control based on its characteristics. Passive fault-tolerant control



Citation: Zhang, Z.; Wu, Y.; Zhou, Y.; Hu, D. Fault-Tolerant Control of Autonomous Underwater Vehicle Actuators Based on Takagi and Sugeno Fuzzy and Pseudo-Inverse Quadratic Programming under Constraints. *Sensors* **2024**, *24*, 3029. https://doi.org/10.3390/s24103029

Academic Editor: Mario Luca Fravolini

Received: 21 March 2024 Revised: 4 May 2024 Accepted: 8 May 2024 Published: 10 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). typically employs a fixed controller structure and parameters, utilizing robust control techniques to render the system insensitive to specific faults [1]. Passive fault-tolerant control is characterized by the ability of the original controller to mitigate the effects of a component failure, ensuring that the system can continue to carry out its tasks with reduced performance. In Ref. [2], a fault-tolerant controller is designed for the depth control of an AUV. The controller utilizes periodic output feedback gains and a multi-model approach. In the event of thruster failure, the multi-model system adjusts by using a gain matrix with all off-diagonal terms set to zero. Ref. [3] proposed a finite-time extended state observer (FTESO)-based scheme for AUV fault-tolerant control problems with multiple-thrusters by estimating the uncertainties caused from faults, modeling uncertainty and system disturbances.

Active fault-tolerant control relies on real-time fault diagnosis information to modify the control inputs of the system to maintain stability. This approach can be classified into two categories based on how the control inputs are adjusted: fault accommodation and control reconfiguration [4]. Control reconfiguration involves the real-time reconstruction of the control system based on fault diagnosis results using specific algorithms. In Ref. [5], genetic algorithms are incorporated into the reconstruction of fault-tolerant control laws to address reliability issues in AUVs. The proposed method utilizes constrained genetic algorithms to reconstruct fault-tolerant control laws for AUVs. By providing relevant fault weight matrices under different fault scenarios, the genetic algorithm is used to search for the optimal solution of the control matrix, facilitating the reconstruction of a propulsion system control matrix. In addressing the actuator fault-tolerant control issue, control reconstruction is accomplished by leveraging the inherent redundant actuators of the robot, thereby enabling the fault-tolerant control of the AUV, as discussed in Refs. [6,7]. A reconfigurable control allocation scheme was proposed in Ref. [8] for possible types of fault in X-rudder UUV, which allows the UUV to operate normally even in cases of partial rudder failure, and its effectiveness was verified through simulation. An observer-based fault-tolerant control scheme for a discrete-time descriptor system with signal-to-noise ratio-constrained channels was proposed in Ref. [9]. In this framework, the augmented robust filter can achieve the simultaneous estimation of system states and faults. The introduction of dynamic thresholds further enhances the performance of this diagnostic method, providing theoretical guidance for the design of fault-tolerant control for AUVs.

Fault regulation involves adjusting the controller parameters or structure based on fault diagnosis information to maintain the input-output relationship between the controller and the controlled system. Ref. [10] developed a continuous adaptive nonlinear model by adjusting the controller or model parameters. In the event of an actuator fault, neural networks and fuzzy logic are employed to modify the model parameters, enabling fault-tolerant control. Refs. [11,12] focused on thrust allocation modeling and fault-tolerant control techniques, utilizing recursive neural networks for remote-controlled AUVs. Ref. [13] established a control energy function using the duality principle, which is based on the thrust allocation model. This approach enables the implementation of a recursive neural-network-based fault-tolerant control algorithm for the thrusters. Ref. [14] proposed a thrust allocation method based on the minimum l2 norm for a specific type of 7000-m manned submersible manufactured by China Shipbuilding Industry Corporation. The correctness and effectiveness of the control allocation algorithm were verified using a hybrid allocation algorithm based on the pseudoinverse matrix and fixed-point allocation on a semi-physical simulation platform. Ref. [15] decoupled fault-tolerant control from a dynamic controller design and used nonlinear programming to minimize allocation errors and control costs as optimization objectives during the control allocation process to solve the fault-tolerant control problem. Ref. [16] proposed a hybrid algorithm based on singular-value decomposition and fixed-point allocation for fault-tolerant control allocation in AUV propulsion systems. Compared to traditional methods, the above-mentioned approach reduces computational complexity by avoiding the problem of pseudoinverse matrix calculation and can satisfy the thruster saturation constraints.

Currently, there is a scarcity of literature on passive fault-tolerant control methods. Reliable stabilization is one of the passive fault-tolerant control methods for controller faults. This concept involves employing multiple compensators in parallel to stabilize the same controlled object [17]. Ref. [18] demonstrated that the essential condition for reliable stabilization with two compensators is the strong stabilizability of the controlled object. Ref. [19] addressed the aforementioned problem by introducing a parameterized approach for designing two dynamic compensators, offering a solution to the reliable stabilization problem. Ref. [20] developed a solution for the reliable stabilization problem by utilizing multiple dynamic compensators for multivariable systems that are not strongly stabilizable. Ref. [21] investigated the integrity concern associated with open circuit faults, proposed a method to solve the static feedback gain matrix, and provided a numerical solution method for the integrity issue of the closed-loop system configuration within actuator faults. Ref. [22] proposed a state feedback law design with actuator fault integrity control, utilizing optimization-solving through numerical solutions of compatible nonlinear equation systems. However, the integrity concern was not entirely resolved. The main challenge lies in the lack of an effective constructive method for solving fault-tolerant control laws, particularly in the study of high-dimensional multivariable systems [23,24]. As a result, this area presents potential research prospects.

This paper presents a fault-tolerant control strategy for AUV actuators, utilizing T-S fuzzy control and pseudoinverse quadratic programming under control constraints. The subsequent sections of this article are structured as follows: Section 2 covers the design of the T-S fuzzy controller. Section 3 delves into the fault-tolerant control of the actuator. Section 4 focuses on the fault-tolerant control of actuators under control constraints. Experimental verification and analyses are shown in Section 5. Conclusions and our future work are presented in Section 6.

#### 2. System Description

# 2.1. T-S Fuzzy Modelling

The underwater dynamics of an AUV are influenced by buoyancy, gravity, thrust, hydrodynamic forces, interference forces, etc. The model in both the static coordinate system and the dynamic coordinate system is illustrated as in Figure 1. Furthermore, the six-degrees-of-freedom motion equations of AUV can be found in to Ref. [25], with detailed definitions of parameters provided in Table 1.



Figure 1. Coordinate diagram for AUV.

Symbols	Descriptions	Units
Х	The force on x axis	Ν
Y	The force on y axis	Ν
Z	The force on z axis	Ν
Κ	The moment on x axis	N·m
Μ	The moment on y axis	N·m
Ν	The moment on z axis	N·m
p,q,r	Roll, pitch and heading angular velocity	rad/s
u, v, w	The velocity on x, y and z axes	m/s
$\varphi,  heta, \psi$	Roll angle, pitch angle, heading angle	rad

Table 1. Coordinate system-related parameter definition for AUV [26].

The AUV underwater motion is a six-degrees-of-freedom spatial motion, which can be represented by linear motion along three axes and rotational motion around three axes in a dynamic coordinate system. Assume that the center of gravity of the AUV (*G*) does not coincide with the origin of the dynamic coordinate system (*O*), and its coordinates in the dynamic coordinate system are  $x_g$ ,  $y_g$ ,  $z_g$ . For the dynamics equation of the AUV, if only the thrust and torque of the actuators are considered, the rigid-body six-degrees-of-freedom

spatial motion equation of the AUV can be represented as follows [25]:

$$\begin{cases} X = m [u - vr + wq - x_g (r^2 + q^2) + y_g (pq - r) + z_g (pr + q)] \\ Y = m [v - ur - wp - y_g (r^2 + p^2) + y_g (qr - p) + x_g (pq + r)] \\ Z = m [w - uq + vp - z_g (p^2 + q^2) + x_g (pr - q) + z_g (qr + p)] \\ K = I_x p + (I_z - I_y) qr + m [y_g (w - uq + vp) - z_g (v + ur - wp)] \\ M = I_x q + (I_x - I_z) pr + m [z_g (u - vr + wq) - x_g (w + vp - uq)] \\ N = I_x r + (I_z - I_y) pq + m [x_g (v - ur + wp) - y_g (u + vr - wq)] \end{cases}$$
(1)

where *m* is the mass of the AUV;  $\rho_A$  is the density of the AUV;  $I_x$ ,  $I_y$  and  $I_z$  are the moments of inertia, which can be represented as follows:

$$\begin{cases} I_{x} = \int_{V} (y^{2} + z^{2}) \rho_{A} dV \\ I_{y} = \int_{V} (x^{2} + z^{2}) \rho_{A} dV \\ I_{z} = \int_{V} (x^{2} + y^{2}) \rho_{A} dV \end{cases}$$
(2)

Furthermore, for AUV actuator fault modeling, this article establishes a parameter deviation model for AUV output thrust and torque, shown as follows:

$$T = \left(T_{n|n|} + b_n\right)n|n| \tag{3}$$

$$M = M_{\delta}(\delta + b_{\delta})u^2 \tag{4}$$

where *T* is the main thrust,  $T_{n|n|}$  is the main thrust coefficient,  $b_n$  is the parameter deviation of main thrust, *n* is the propeller speed, *M* is the thrust torque,  $M_{\delta}$  is the rudder angle coefficient,  $\delta$  is the rudder angle, and  $b_{\delta}$  is the rudder angle deviation. The propeller and rudder are operating normally when  $b_n = 0$  and  $b_{\delta} = 0$ . Applying the aforementioned actuator fault description to the AUV nonlinear dynamics, one can obtain the following:

$$m(\dot{u} - vr + wq) = X_{u|u|}u|u| + X_{\dot{u}}\dot{u} + X_{vr}vr$$
  
-  $(T_{n|n|} + b_{Ln})n_L|n_L| + (T_{n|n|} + b_{Rn})n_R|n_R|$  (5)

$$m(\dot{v}+ur) = Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_{ur}ur + Y_{uw}uw + Y_{\delta}(\delta+b_{\delta})u^2$$
(6)

$$I_{z}\dot{r} = N_{v|v|}v|v| + N_{r|r|}r|r| + N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} + N_{ur}ur + N_{uv}uv + N_{\delta}(\delta + b_{\delta})u^{2} + \left(\left(T_{n|n|} + b_{Ln}\right)n_{L}|n_{L}| - \left(T_{n|n|} + b_{Rn}\right)n_{R}|n_{R}|\right)d$$
(7)

where the distance *d* is the distance between the left and right thrusters and the longitudinal centerline of the AUV.

To address the issue of fault-tolerant control allocation for the AUV actuator, it is essential to provide a detailed description of the mathematical models involved.

The AUV considered in this paper is a small, torpedo-shaped AUV. Considering the modeling of its actuators, the following form of the nonlinear dynamic system is considered:

$$\begin{cases} \dot{x}(t) = f(x(t)) + g(x(t))u(t) \\ y(t) = Cx(t) \end{cases}$$
(8)

where  $x(t) = [u \ v \ w \ q \ r]^T$  is the state vector of the system; u(t) is the control input to the system;  $f(\cdot)$  is the nonlinear term of the system; g(x(t)) is the control input matrix for the system; y(t) is the output of the system; C is the known constant coefficient matrix.

We initially constructed nonlinear T-S fuzzy models for the AUV control system, based on which the state feedback-based fuzzy controllers utilizing these models were developed. The T-S fuzzy model aims to approximate nonlinear systems by employing multiple local linear models established according to the IF-THEN rules. These rules effectively transform intricate nonlinear problems into familiar linear ones [27]. Each rule corresponds to a subsystem within a nonlinear system that illustrates the dynamic characteristics within each local region. Global nonlinearity is attained through fuzzy inference, derived from local linearization.

For the control system of AUV (8), the fuzzy rules are formulated as follows:

Rule 
$$i$$
: IF  $\theta_1(t)$  is  $M_1^i$  and  $\cdots$  and  $\theta_r(t)$  is  $M_r^i$ ;

Then

$$\begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \ i = 1, 2, \cdots, r \\ y(t) = C x(t) \end{cases}$$
(9)

where  $M_1^i, M_2^i, \dots, M_r^i$  are fuzzy sets;  $A_i, B_i, C_i$  are known real number matrices with corresponding dimensions;  $\theta_1(t), \theta_2(t), \dots, \theta_r(t)$  are known prior variables, which can be a state variable, external disturbance, a time function, or a combination of these vectors.

By integrating the individual subsystems using methods such as single-point fuzzification, product inference, and weighted average defuzzification, we can derive the ultimate hybrid model of the nonlinear system, as illustrated below. The state equation of the entire fuzzy system can be expressed as follows:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{r} \mu_i(\theta(t)) (A_i x(t) + B_i u(t)) \\ y(t) = \sum_{i=1}^{r} \mu_i(\theta(t)) C x(t) \end{cases}$$
(10)

where  $\mu_i(\theta) = \frac{\omega_i(\theta(t))}{\sum_{j=1}^r \omega_i(\theta(t))}$  is the membership function,  $\omega_i(\theta(t)) = \prod_{j=1}^s M_j^i(\theta_j)$ . The following conditions are satisfied:

$$\mu_i(\theta(t)) > 0, \sum_{i=1}^r \mu_i(\theta(t)) = 1$$
(11)

The importance of constructing a fuzzy model lies in its ability to approximate a nonlinear dynamic model as an ensemble of local linear models, thereby enhancing fuzzy precision by incorporating a larger number of fuzzy rules. However, as the number of fuzzy rules increases, the design intricacy of fuzzy controllers also increases. Therefore, system modeling must strike a balance between accuracy and complexity.

In order to better elucidate the fault-tolerant control scheme proposed in this article, the flowchart shown as in Figure 2 is provided. Specific steps will be detailed in subsequent chapters.



Figure 2. The flowchart of fault-tolerant control scheme.

## 2.2. The Design and Stability of Fuzzy Controllers

When designing controllers for T-S fuzzy models, the prevalent approach in the current research involves the utilization of the parallel distribution compensation method. In the controller design, the premise variables of each controller's fuzzy rule are the same as the corresponding premise variables of the fuzzy model. Subsequently, a state feedback control law is formulated for each rule's respective linear subsystem, followed by the derivation of the control law for the global system through fuzzification weighting.

For the T-S fuzzy system in Section 2.1, the state feedback controller is structured based on the parallel distribution compensation algorithm. The controller rules are outlined as follows:

Rule 
$$i$$
: IF  $\theta_1(t)$  is  $M_1^i$  and  $\cdots$  and  $\theta_r(t)$  is  $M_r^i$ ;

Then

$$u(t) = K_j x(t) \ j = 1, 2, \cdots, r$$
 (12)

where  $K_j$  is the gain matrix for the compensation of the controller distribution that needs to be designed. By amalgamating the controllers of the aforementioned subsystems, the comprehensive fuzzy controller can be acquired, as depicted below:

$$u(t) = \sum_{j=1}^{r} \mu_j(\theta(t)) K_j x(t)$$
(13)

Furthermore, a fuzzy model of the closed-loop system is obtained, as follows:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{r} \sum_{j=1}^{r} \mu_i(\theta(t)) \mu_j(\theta(t)) \left(A_i + B_i K_j\right) x(t) \\ y(t) = \sum_{i=1}^{r} \mu_i(\theta(t)) C x(t) \end{cases}$$
(14)

When designing a system, the stability of the control system should be prioritized to guarantee the smooth operation of the system. To establish a robust stability criterion for the fuzzy control system (14), a controller can be designed from a theoretical standpoint.

Take the Lyapunov function as  $V(x(t)) = [x(t)]^T P x(t)$ , where *P* is a symmetric and positive-definite matrix; then, one can obtain the following:

$$\dot{V}(x(t)) = [\dot{x}(t)]^{T} P x(t) + [x(t)]^{T} P \dot{x}(t) 
= \left[\sum_{i=1}^{r} \sum_{j=1}^{r} \mu_{i}(\theta(t)) \mu_{j}(\theta(t)) (A_{i} + B_{i}K_{j}) x(t)\right]^{T} P x(t) 
+ [x(t)]^{T} P \left[\sum_{i=1}^{r} \sum_{j=1}^{r} \mu_{i}(\theta(t)) \mu_{j}(\theta(t)) (A_{i} + B_{i}K_{j}) x(t)\right]$$

$$= \sum_{i=1}^{r} \sum_{j=1}^{r} \mu_{i}(\theta(t)) \mu_{j}(\theta(t)) [x(t)]^{T} [(A_{i} + B_{i}K_{j})^{T} P + P(A_{i} + B_{i}K_{j})] x(t)$$
(15)

For the formula (15), the Left and right side of  $(A_i + B_i K_i)^T P + P(A_i + B_i K_j)$  are multiplied by  $P^{-1}$  as  $P^{-1}(A_i + B_i K_j)^T + (A_i + B_i K_j)P^{-1}$ . Denote  $H_i = K_i P^{-1}$ 

$$Q_{ij} = \left(A_i R + B_i H_j\right)^T + \left(A_i R + B_i H_j\right) \tag{16}$$

The variables in the Equation (16) must satisfy the following conditions:

$$\begin{cases} Q_{ij} < 0 \ i = 1, 2, \cdots, r \\ \frac{1}{r-1}Q_{ij} + 0.5(Q_{ij} + Q_{ji}) < 0 \ 1 \le i \ne j \le r \end{cases}$$
(17)

When the aforementioned conditions are fulfilled,  $\dot{V}(x(t)) < 0$ , then the T-S fuzzy system exhibits asymptotic stability.

#### 2.3. The Design of T-S Fuzzy Rules

The variations in the state variables v, w, q and r in the six-degrees-of-freedom motion equations of an AUV are interconnected with the linear velocity u. Changes in u will impact the velocities and angular velocities of other degrees of freedom. Additionally, there is a constraint relationship between q and r in the posture equation, as both are influenced by the control moments simultaneously. To facilitate future research, it is essential to simplify the variables needed for the fuzzy controller design. Denote  $\theta_1 = u, \theta_2 = r$ ; choose  $\theta_1$ and  $\theta_2$  as the premise variables of the T-S fuzzy system. According to the actual situation, the corresponding fuzzy sets are designed as follows: { $\theta_1 = 0.5, 1$ }, { $\theta_2 = -0.1, 0, 0.1$ }.

The commonly used membership function information includes the upper and lower bounds of the membership function, as well as boundary information. For instance, one can approximate the membership function using a ladder function or by selecting a certain number of equally spaced points and approximating the information with adjacent points. Generally, the membership function in a fuzzy model typically mirrors that of a fuzzy controller, requiring the consideration of accuracy requirements and computational complexity. In the context of the actual AUV system, we select the following membership functions for the AUV actuator fuzzy controller:

$$M_{\theta_1=0.5} = \frac{2 + \sin(\theta_1(t))}{5}, M_{\theta_1=1} = \frac{3 - \sin(\theta_1(t))}{5}$$
(18)

$$M_{\theta_2=-0.1} = \frac{\cos(\theta_2(t)) + 2}{5}, M_{\theta_2=0} = \frac{\sin(\theta_2(t)) + 3}{5}, M_{\theta_2=0.1} = \frac{-\sin(\theta_2(t)) - \cos(\theta_2(t)) + 5}{5}$$
(19)

For Equations (18) and (19), six working points are selected with reference to the fuzzy set [0.5, -0.1], [0.5, 0], [0.5, 0.1], [1, -0.1], [1, 0], [1, -0.1]. Thus, the design of AUV fuzzy model and fuzzy controller are as follows:

Rule1: IF  $\theta_1$  is about 0.5 m/s and  $\theta_2$  is about -0.1 rad/s, then

$$\begin{cases} \dot{x}(t) = A_1 x(t) + B_1 u(t) \\ u(t) = K_1 x(t) \end{cases}$$
(20)

Rule2: IF  $\theta_1$  is about 0.5 m/s and  $\theta_2$  is about 0 rad/s, then

$$\begin{cases} \dot{x}(t) = A_2 x(t) + B_2 u(t) \\ u(t) = K_2 x(t) \end{cases}$$
(21)

Rule3: IF  $\theta_1$  is about 0.5 m/s and  $\theta_2$  is about 0.1 rad/s, then

$$\begin{cases} \dot{x}(t) = A_3 x(t) + B_3 u(t) \\ u(t) = K_3 x(t) \end{cases}$$
(22)

Rule4: IF  $\theta_1$  is about 1 m/s and  $\theta_2$  is about -0.1 rad/s, then

$$\begin{cases} \dot{x}(t) = A_4 x(t) + B_4 u(t) \\ u(t) = K_4 x(t) \end{cases}$$
(23)

Rule5: IF  $\theta_1$  is about 1 m/s and  $\theta_2$  is about 0 rad/s, then

$$\begin{cases} \dot{x}(t) = A_5 x(t) + B_5 u(t) \\ u(t) = K_5 x(t) \end{cases}$$
(24)

Rule6: IF  $\theta_1$  is about 1 m/s and  $\theta_2$  is about 0.1 rad/s, then

$$\begin{cases} \dot{x}(t) = A_6 x(t) + B_6 u(t) \\ u(t) = K_6 x(t) \end{cases}$$
(25)

In the above rules, the controller distribution compensation gain vector of each subsystem is as follows:  $K = [K_1 \ K_2 \ K_3 \ K_4 \ K_5 \ K_6]$ . By combining the known membership functions, a complete fuzzy controller can be designed to provide control input for the AUV motions and provide the premise for the subsequent fault-tolerance control.

## 3. Fault-Tolerant Control of Auv Actuators

# 3.1. Arrangement and Normalization Strategy of the Propulsion System

The devices collectively powering the AUV are known as its propulsion system, which includes propellers and rudder engines (as shown in Figure 3). The propellers consist of two horizontal thrusters and two vertical thrusters, producing thrusts T1, T2, T3, T4 (actually, T1 and T2 are the same, described as *u*1; T3 and T4 are the same, described as *u*2), located at the tail of the AUV. The rudder engines consist of vertically and horizontally distributed rudders that produce thrust moments T5, T6, T7, T8 (where T5 and T6 are denoted as *u*3, and T7 and T8 are represented as *u*4). The propulsion system of the AUV has the ability to directly control all six degrees of freedom, making it equipped with redundant propulsion control. In the event of a specific thruster or rudder failing completely or experiencing partial malfunction, the lost control effect can be redistributed by other thrusters or rudders



based on predetermined criteria. This redistribution enables the fault-tolerant control of the AUV propulsion system.

Figure 3. Propulsion system layout for AUV.

Based on the above analysis, during the operation of an AUV, the forces and moments at a specific moment can be represented using the state vector  $\tau$ , where  $\tau = [\tau_x \tau_y \tau_z]$  is the thrust force, and  $\tau = [\tau_m \tau_n]$  is the moment generated by the control signal provided by the entire propulsion system. For the AUV control system, each state corresponds to a specific motion state, with forces and moments generated by the control signal *u*. In the context of AUV fault-tolerant control, the goal is to reconfigure a new set of thruster control signals based on the arrangement of the thrusters in the event of actuator faults. This reconfiguration ensures that the thruster can still provide sufficient force and moments to maintain control and stability of the AUV [28].

For the propulsion system arrangement described above, the force and moments formula is as follows:

$$\tau = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \\ \tau_m \\ \tau_n \end{bmatrix} = \begin{bmatrix} l\cos\alpha & \cos\alpha & \cos\beta & \cos\beta \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & \cos\beta & \sin\beta \\ A & A & A & A \\ B & B & B & B \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = B \cdot u$$
(26)

where  $A = \frac{b}{2} \sin \alpha + \frac{a}{2} \cos \alpha$ ,  $B = \frac{b}{2} \cos \beta + \frac{a}{2} \sin \beta$ .

In order to more intuitively reflect the utilization of thrust from each thruster and to prevent potential thrust saturation output when the multiple desired control quantities are simultaneously output, a detailed description of the normalization process for the output resultant thrust of the propulsion system is necessary. Assume the maximum control vector provided by the thruster can be represented as follows:

$$\begin{cases} \tau_{xH} = 2u_m \cos \alpha + 2u_m \cos \beta = 4u_m \cos \alpha = 4u_m \cos \beta \\ \tau_{yH} = u_m \sin \alpha + u_m \cos \alpha \\ \tau_{zH} = u_m \sin \beta + u_m \cos \beta \\ \tau_{mH} = 4A \\ \tau_{nH} = 4B \end{cases}$$
(27)

By performing a conversion of the above equation, one can obtain the following:

$$\begin{aligned}
\bar{\tau}_x &= \frac{\tau_x}{u_m} = \frac{1}{4\cos\alpha} = \frac{1}{4}\cos\beta \\
\bar{\tau}_y &= \frac{\tau_y}{u_m} = \frac{1}{2\sin\alpha} \\
\bar{\tau}_z &= \frac{\tau_z}{u_m} = \frac{1}{2\sin\beta} \\
\bar{\tau}_m &= \frac{\tau_m}{u_m} = \frac{1}{2}A \\
\bar{\tau}_n &= \frac{\tau_n}{u_m} = \frac{1}{2B}
\end{aligned}$$
(28)

The final expression for the normalized state vector can be obtained as follows:

$$\bar{\tau} = \begin{bmatrix} \bar{\tau}_{x} \\ \bar{\tau}_{y} \\ \bar{\tau}_{z} \\ \bar{\tau}_{m} \\ \bar{\tau}_{n} \end{bmatrix} = \begin{bmatrix} l\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} \bar{u}_{1} \\ \bar{u}_{2} \\ \bar{u}_{3} \\ \bar{u}_{4} \end{bmatrix} = \bar{B}\bar{u}$$
(29)

For Equation (29), when the thrusters are operating normally, a certain thruster control voltage u will generate a corresponding rotational speed n, and then obtain the corresponding thrust and torque. When a fault occurs in the propulsion system, for the same control voltage, the output rotational speed n will decrease due to the presence of the fault, resulting in an inability to obtain the corresponding thrust and torque.

At this point, the thruster speed n can more accurately reflect the thrust and torque of the thruster during actual operation. In general, the relationship between the thrust and torque of a thruster and its rotational speed is a complex, non-linear one. However, for AUVs, this can often be approximated as a linear relationship. Therefore, Equation (31) can be used to calculate the actual force state of the AUV and compare it with the desired force state to evaluate the fault-tolerant control effectiveness.

$$\bar{\tau} = \begin{bmatrix} \bar{\tau}_x \\ \bar{\tau}_y \\ \bar{\tau}_z \\ \bar{\tau}_m \\ \bar{\tau}_n \end{bmatrix} = \begin{bmatrix} l_{\frac{1}{4}} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} \bar{n}_1 \\ \bar{n}_2 \\ \bar{n}_3 \\ \bar{n}_4 \end{bmatrix} = \bar{B}\bar{n}$$
(30)

#### 3.2. Pseudo-Inverse-Based Fault-Tolerant Control

The redundant configuration of the propulsion system in Section 2.2 allows for the fault-tolerant control of the thrusters. By utilizing the thruster priority matrix W, the control priority of each thruster can be expressed as follows:

$$W = \begin{bmatrix} w_1 & 0 & 0 & 0\\ 0 & w_2 & 0 & 0\\ 0 & 0 & w_3 & 0\\ 0 & 0 & 0 & w_4 \end{bmatrix}$$
(31)

In fault diagnosis and the fault-tolerant control of AUVs, the thruster priority matrix W is closely related to the thruster faults. When none of the structures in the propulsion system have failed, the priority of each thruster signal u is equal, i.e.,  $w_1 = w_2 = w_3 = w_4 = 1$ . In the event of a thruster fault,  $f_i$  is considered as a fault factor. If the *i*-th thruster experiences complete failure,  $f_i = 0$ . If the corresponding thruster experiences partial failure, then  $0 < f_i < 1$ .

Furthermore, the priority coefficients of partially failed thrusters can be represented as follows:

$$w_i = e^{\frac{1}{f} - 1} \tag{32}$$

If the thruster fails completely,  $w_i = e^{+\infty} = +\infty$ , then

$$f_i = n_i (1 - s_i) \tag{33}$$

where  $n_i$  is the normalized thruster speed;  $s_i$  is the congestion parameter.  $0 < s_i < 1$  depends on the severity of the fault. If the thruster completely fails,  $s_i = 0$ . For partial thruster faults, such as  $s_i = 0.5$ , the output operating range of the thruster is limited to 50.

In this scenario, the fault-tolerant ability of the AUV system is analyzed in the event that a specific thruster encounters a time-varying fault. The affected thruster will experience a partial loss of driving function. The goal of the control system is to develop a new set of control signals to replace the original four thruster control signals that included the fault signal included in the fault occurrence. This ensures that the thruster can continue to generate the required thrust and torque even in the event of a fault, maintaining the AUV's original motion states.

Assuming that the control signals of the thruster can meet the specified control quantity, the pseudoinverse allocation solution can be obtained using the Lagrange multiplier algorithm [29]. The Lagrangian function is defined as follows:

$$L(u,\lambda) = \frac{1}{2}u^T W u + \lambda(\tau - Bu)$$
(34)

where  $\lambda$  is a Lagrange multiplier. By taking the derivative of it, one can obtain:

$$\frac{\partial L}{\partial u}Wu - B^T\lambda = 0 \tag{35}$$

Therefore, we can obtain:

$$\tau = Bu = BW^{-1}B^T\lambda \tag{36}$$

Finally, the solution for pseudoinverse control allocation is obtained as follows:

$$T = \bar{B}^+ u = \left[ W^{-1} \bar{B}^T \left( \bar{B} W^{-1} \bar{B}^T \right) \right] u \tag{37}$$

where  $\bar{B}^+$  is the pseudoinverse weight matrix of the thruster configuration matrix  $\bar{B}$ .

#### 4. Fault-Tolerant Control of Actuators under Control Constraints

4.1. Description of Control Constraints

Section 3 studies the thrust allocation control using the pseudo-inverse method to ensure the normal operation of the AUV under ideal conditions. However, in practical operations, there are constraints on the input controls that drive the motion of the AUV. When the control inputs exceed the constraint limits, the pseudo-inverse method cannot automatically adjust the remaining control inputs to compensate for the insufficient control force caused by the constraints, leading to the actual control force failing to achieve the desired control effect.

Due to the influence of its own dynamics and water flow, the propulsion system of an AUV has limitations and saturation nonlinearity in the forces and torques. Figure 4 illustrates the typical input saturation characteristics of the propulsion system.



Figure 4. Schematic diagram of input saturation characteristics.

The mathematical expression is described as follows:

$$u_{in} = \begin{cases} u_{\max} & u_c > u_{\max} \\ k u_c & u_{\min} < u_c < u_{\max} \\ u_{\min} & u_c < u_{\min} \end{cases}$$
(38)

When implementing fault-tolerant control in the presence of actuator faults, it is crucial to consider saturation constraints. In practical design, if the required control inputs exceed the range that the propulsion system can provide, the controller will continuously accumulate errors, resulting in system instability and potentially causing damage to the actuators.

Suppose that the insufficient control force and torque can be redistributed through the reallocation of the unsaturated control signals. The deviation matrix of the actuator control signal is denoted as M. If the maximum and minimum control input vectors are  $T_{\text{max}}$ ,  $T_{\text{min}}$ , the feasible range of the actuator control signal during reallocation is determined as follows:

$$\begin{cases}
M \le T_{\max} - T \\
M \ge T - T_{\min}
\end{cases}$$
(39)

and B(T+M) = u.

According to the properties of matrix multiplication, one can obtain:

$$\begin{cases}
BM = u - BT \\
BM = O_{m \times 1}
\end{cases}$$
(40)

To ensure that the L2-norm of the control signals of the redistributed propulsion system is minimized, it is necessary to ensure that the L2-norm of the control signals obtained from the two control allocations is minimized.

According to the norm properties, one can obtain:

$$||T + M|| \le ||T|| + ||M|| \tag{41}$$

Furthermore, the problem is transformed into solving min $||M_{dev}||$ . Finally, the aforementioned problem can be formulated in the following standard form of quadratic programming:

min 
$$J = M^T H M + C M$$
  
s.t.  $A_i M + B_i = 0, i \in E$   
 $A_i M + B_i \ge 0, i \in I$ 
(42)

where *H* is the Hessian matrix; *C* is the coefficient matrix of the linear term for the variable M;  $A_i$ ,  $B_i$  are the coefficient matrices of the linear terms and constant terms in the constraint functions for the variable M, respectively.

#### 4.2. Nonlinear Quadratic Programming

The nonlinear quadratic programming algorithm is efficient in solving nonlinear programming problems. Compared to other algorithms, its most prominent advantages include its good convergence, high computational efficiency, and strong boundary search capabilities. It has been widely used in many fields. All non-linear programming models are generally composed of an optimization objective function and corresponding constraints. For optimization problems with inequality constraints, this paper uses the active set method to solve quadratic programming, shown as follows.

Assuming  $\overline{M}$  is a feasible point for Equation (44), its active set can be defined as the set of constraints for which the equality holds, that is:

$$A(\bar{M}) = \{ i \in (E \cup I) : A_i \bar{M} = B_i \}$$
(43)

For quadratic programming problems with inequality constraints, the Lagrangian function can be described as follows:

$$L(M,\lambda) = M^T H M + C M - \sum_{i \in (E \cup I)} \left[\lambda_i (A_i \bar{M} - B_i = 0)\right]$$

$$\tag{44}$$

(45)

Furthermore, the optimality condition (K-T) for quadratic programming can be described as follows:  $H\bar{M} + C - \sum_{i \in A(\bar{M})} \lambda_i A_i = 0$ 

$$A_{i}\bar{M} = B_{i}, i \in A(\bar{M})$$

$$A_{i}\bar{M} \ge B_{i}, i \in I \cup A(\bar{M})$$

$$\lambda_{i} \ge 0, i \in I \cap A(\bar{M})$$
(46)

and  $\overline{M}$  is also referred to as the K-T point of Equation (46).

The active set method is essentially an iterative algorithm that uses the feasible point of Equation (46) as the starting point in each iteration and defines the search direction with the set of active constraints, then determines the search step size through linear search.

Let  $M_k$  be the feasible solution at the k-th step, with its active set denoted as  $S_k = E \cup I(M_k)$ ; consider the following optimization problem:

$$\min\left(M^{T}HM + C^{T}M\right)$$
  
s.t.  $A_{i}M = B_{i}, i \in S_{k}$  (47)

Denote the step length  $p = M - M_k$  and substituting it into Equation (47) while removing the constant term yields the equivalent optimization problem, as follows:

$$\min\left(p^{T}Hp + p_{k}^{T}p\right)$$
  
s.t.  $A_{i}p = B_{i}, i \in S_{k}$  (48)

where  $p_k = p + HM_k$ . Find the optimal solution  $p_k$  of quadratic programming under equality constraints and divide it into the following three cases:

(1)  $p_k \neq 0$ , while  $M_k + p_k$  is a feasible point of Equation (47). Take a new iteration point  $M_{k+1} = M_k + p_k$  and then calculate the corresponding optimization problem.

(2)  $M_k + p_k$  is a feasible point of Equation (47). Denote  $M_{k+1} = M_k + \alpha_k p_k$ , where the step size parameter  $\alpha_k \in (0, 1]$  can be represented in detail as follows to ensure that  $M_{k+1}$ in (2) is a feasible point.

$$\alpha_k = \min\left\{1, \min\frac{B_i - B_i M_k}{A_i p_k}\right\}$$
(49)

(3)  $p_k = 0$ ,  $M_k$  is the optimal solution of the optimization problem (48). Determining  $M_k$  in (3) is the K-T point of the original problem. If all Lagrange multipliers of the original problem are non-negative, it can be concluded that this point is the optimal solution of the quadratic programming problem (47) based on the convex optimization theory.

Quadratic programming can be used to optimize and solve the problems of minimizing tracking errors and control inputs. When the propulsion system fails, quadratic programming can dynamically update actuator matrices based on information, achieving re-allocation under control constraints.

#### 5. Experimental Verification and Analysis

#### 5.1. Simulation Settings

In this section, simulations are conducted for the actuator fault-tolerant control of an AUV with known fault types. The faults are set as follows: (1) Thruster fault: at 150 s, one of the thrusters experiences a thrust degradation to 0, while the parameters of the other thrusters remain unchanged; (2) Rudder servo fault: at 150 s, one of the servos fails, with its torque degrading to 0, rendering it unable to deflect and provide the designed torque, while the parameters of the other servos remain unchanged. It should be noted that

the two types of fault conditions mentioned in this article will not occur simultaneously and are not related to each other.

#### 5.2. Simulation Results and Analysis

(1) Fuzzy controller input signals

Figure 5 illustrates the control input signals generated by the T-S fuzzy controller for the actuator of the AUV, which are provided by the thrusters and the rudder servo. From the figure, it can be observed that the input control signals generated by the fuzzy controller are stable, indicating that the actuators of the AUV can provide a stable thrust and torque to the AUV.



Figure 5. Input signal for fuzzy controller.

(2) Pseudo-inverse-based fault-tolerant control (without control constraints)

Figures 6 and 7 show the pseudo-inverse fault-tolerant control based on thrust allocation. It can be observed that when the control inputs of the AUV actuator degrade due to faults, the output states change accordingly. By employing the pseudo-inverse method for the thrust allocation of faulty thrusters and rudders, the lost thrust and torque can be effectively compensated, enabling the AUV to maintain a certain level of system dynamic performance even in the presence of faults.



Figure 6. Fault-tolerant control under thrust loss (abrupt fault + slowly-varying fault).

(3) Quadratic programming under control constraints

Figures 8 and 9 illustrate the thrust allocation results based on pseudo-inverse control under control constraints. When the control input fails to provide the ideal signal due to various factors, the pre-designed control signals exceed the constraint range. The original fault-tolerant thrust allocation method is unable to fully compensate for the missing effectiveness due to faults, resulting in the originally designed fault-tolerant control scheme failing to meet the expected effectiveness, which leads to significant deficiencies in practical applications.

Figures 10 and 11 present the fault-tolerant control results of the AUV actuators, combined with pseudo-inverse and nonlinear quadratic programming. The quadratic programming algorithm compensates for the missing control input within control constraints, allowing the input thrust of the AUV actuator to return to normal and achieving the expected fault-tolerant effect



Figure 7. Fault-tolerant control under moment loss (abrupt fault + slowly-varying fault).



Figure 8. Fault-tolerant control under control constraints (abrupt fault).



Figure 9. Fault-tolerant control under control constraints (slowly varying fault).



Figure 10. Fault-tolerant control under quadratic programming (abrupt fault).

In general, the T-S fuzzy logic and pseudo-inverse quadratic programming-based faulttolerant scheme in this paper can resolve the issue of AUV actuator faults under control constraints. However, in terms of fault-tolerant controller design, the proposed solution in this paper does not consider the requirement for robustness. Inspired by Ref. [30], sliding mode control (SMC) exhibits strong robustness to external disturbances and parameter uncertainties, with a good response and tracking performance, making it suitable for scenarios with high demands on system dynamic performance. Therefore, as shown in Figures 12 and 13, the SMC approach can be taken into consideration in the design of fault-tolerant controllers for both abrupt faults and slowly varying faults based on the previous quadratic programming. Furthermore, based on the comparisons in Figures 10–13 (as indicated by the solid red line), the fault-tolerant control obtained under quadratic programming and SMC design can obtain advanced robustness without sacrificing the nominal control performance of AUVs.



Figure 11. Fault-tolerant control under quadratic programming (slowly-varying fault).



Figure 12. Fault-tolerant control under quadratic programming and SMC (abrupt fault).



Figure 13. Fault-tolerant control under quadratic programming and SMC (slowly-varying fault).

# 6. Conclusions

In this paper, we propose a fault-tolerant control method for AUV actuators based on T-S fuzzy logic and pseudo-inverse quadratic programming, taking into account control

constraints in practical scenarios. Additionally, the thrust allocation method, employing pseudo-inverse quadratic programming, has demonstrated effective fault-tolerant control for AUV actuators. The simulation results indicate that the proposed fault-tolerant control method with control constraints can effectively compensate for missing control input signals, ensuring the normal operation of AUVs in the event of actuator faults. Our future work will focus on efficient fault diagnosis methods [31–33] to obtain comprehensive fault information for AUVs to further improve fault tolerance, and the data from real-world experiments (not only the numerical simulations) will be used to demonstrate the controller's performance in practical scenarios.

**Author Contributions:** Conceptualization, Z.Z. and Y.W.; methodology, Y.W. and Y.Z.; software, D.H.; validation, Z.Z. and Y.W.; formal analysis, Z.Z.; investigation, Y.W.; resources, Y.Z.; data curation, Y.W. and D.H.; writing—original draft preparation, Z.Z. and D.H.; writing—review and editing, Y.W.; visualization, Y.Z.; supervision, Y.Z.; project administration, Y.W.; funding acquisition, Y.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grants (62173164, 62203192), in part by the 6th regular meeting exchange program (6-3) of the China-North Macedonian science & technology cooperation committee.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

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

### Abbreviations

The following abbreviations are used in this manuscript:

AUV	Autonomous Underwater Vehicle
T-S	Takagi and Sugeno
FTESO	Finite-Time Extended State Observer

#### References

- Kurtoglu, D.; Bidikli, B.; Tatlicioglu, E.; Zergeroglu, E. Periodic disturbance estimation based adaptive robust control of marine vehicles. Ocean Eng. 2020, 219, 108351. [Crossref] [CrossRef]
- 2. Joshi, S.; Talange, B.D. Fault Tolerant Control of an AUV using Periodic Output Feedback with Multi Model Approach. *Int. J. Syst. Dyn. Appl.* (*IJSDA*) **2016**, *5*, 41–62. [Crossref] [CrossRef]
- 3. Liu, X.; Zhang, M.; Liu, X.; Zhao, W. Finite-Time Extended State Observe Based Fault Tolerant Control for Autonomous Underwater Vehicle with Unknown Thruster Fault. J. Mar. Sci. Eng. 2022, 10, 1624. [Crossref] [CrossRef]
- He, S.; Xu, Z.; Li, Y.; Xiang, J. Fault diagnosis and reconfigurable control for underwater vehicles. *Ocean Eng.* 2021, 239, 109813. [Crossref] [CrossRef]
- 5. Huang, C.; Naghdy, F.; Du, H.P. Delta operator-based fault estimation and fault-tolerant model predictive control for steer-by-wire systems. *IEEE Trans. Control. Syst. Technol.* **2017**, *26*, 1810–1817. [Crossref] [CrossRef]
- Jin, Y.; Zhang, Y.M.; Jing, Y.W. An average dwell-time method for fault-tolerant control of switched time-delay systems and its application. *IEEE Trans. Ind. Electron.* 2018, 66, 3139–3147. [Crossref] [CrossRef]
- 7. Zhang, S.; Yang, P.X.; Kong, L.H.; 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] [CrossRef]
- Xu, H.; Feng, J.; Zhang, P.; Ma, C. Research on Fault-tolerant Control Allocation Technology for X-rudder UUV. J. Phys. Conf. Ser. 2023, 2674, 012037. [Crossref] [CrossRef]
- 9. Guo, F.; Ren, X.; Li, Z.; Han, C. Fault detection and fault-tolerant control over signal-to-noise ratio constrained channels. *Int. J. Robust Nonlinear Control.* 2018, *28*, 2806–2827. [Crossref] [CrossRef]
- 10. Ding, R.Q.; Cheng, M.; Jiang, L.; Hu, G. Active fault-tolerant control for electro-hydraulic systems with an independent metering valve against valve faults. *IEEE Trans. Ind. Electron.* **2020**, *68*, 7221–7232. [Crossref] [CrossRef]
- 11. Zhang, L.; Wang, Z.H.; Ding, X.L.; Li, S.; Wang, Z. Fault-tolerant control for intelligent electrified vehicles against front wheel steering angle sensor faults during trajectory tracking. *IEEE Access* **2021**, *9*, 65174–65186. [CrossRef] [CrossRef]

- Xiao, Q.; Chen, L.L.; Jin, Y.; Mu, Y.; Cupertino, A.F.; Jia, H.; Neyshabouri, Y.; Dragicevic, T.; Teodorescu, R. An improved fault-tolerant control scheme for cascaded H-bridge STATCOM with higher attainable balanced line-to-line voltages. *IEEE Trans. Ind. Electron.* 2020, *68*, 2784–2797. [Crossref] [CrossRef]
- Amin, A.A.; Mahmood, U.; Hasan, K. Advanced fault tolerant air-fuel ratio control of internal combustion gas engine for sensor and actuator faults. *IEEE Access* 2019, 7, 17634–17643. [Crossref] [CrossRef]
- 14. Khan, S.S.; Wen, H.Q. A comprehensive review of fault diagnosis and tolerant control in DC-DC converters for DC microgrids. *IEEE Access* 2021, 9, 80100–80127. [Crossref] [CrossRef]
- 15. Yu, X.; Li, P.; Zhang, Y.M. The design of fixed-time observer and finite-time fault-tolerant control for hypersonic gliding vehicles. *IEEE Trans. Ind. Electron.* **2017**, *65*, 4135–4144. [Crossref] [CrossRef]
- 16. Gao, M.Z.; Cai, G.P. Finite-time H∞ adaptive fault-tolerant control for wing flutter of reentry vehicle subject to input saturation. *Int. J. Control. Autom. Syst.* 2017, 15, 362–374. [Crossref] [CrossRef]
- 17. Wang, T.; Gao, H.; Qiu, J.B. A combined fault-tolerant and predictive control for network-based industrial processes. *IEEE Trans. Ind. Electron.* **2016**, *63*, 2529–2536. [Crossref] [CrossRef]
- Wang, J.; Zhang, J.; Qu, B.; Wu, H.; Zhou, J. Unified architecture of active fault detection and partial active fault-tolerant control for incipient faults. *IEEE Trans. Syst. Man Cybern. Syst.* 2017, *47*, 1688–1700. [Crossref] [CrossRef]
- 19. Tian, Q.; Wang, T.; Liu, B.; Ran, G. Thruster Fault Diagnostics and Fault Tolerant Control for Autonomous Underwater Vehicle with Ocean Currents. *Machines* 2022, 10, 582. [Crossref] [CrossRef]
- 20. Wang, Y.; Song, Y.D.; Lewis, F.L. Robust adaptive fault-tolerant control of multiagent systems with uncertain nonidentical dynamics and undetectable actuation failures. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3978–3988. [Crossref]
- 21. Arafat, A.K.M.; Choi, S. Optimal phase advance under fault-tolerant control of a five-phase permanent magnet assisted synchronous reluctance motor. *IEEE Trans. Ind. Electron.* 2017, *65*, 2915–2924. [CrossRef] [CrossRef]
- Xu, S.D.; Chen, C.C.; Wu, Z.L. Study of nonsingular fast terminal sliding-mode fault-tolerant control. *IEEE Trans. Ind. Electron.* 2015, 62, 3906–3913. [Crossref] [CrossRef]
- 23. Shen, Q.; Yue, C.F.; Goh, C.H.; Wang, D. Active fault-tolerant control system design for spacecraft attitude maneuvers with actuator saturation and faults. *IEEE Trans. Ind. Electron.* **2018**, *66*, 3763–3772. [Crossref] [CrossRef]
- Yin, S.; Gao, H.; Qiu, J.B.; Kaynak, O. Adaptive fault-tolerant control for nonlinear system with unknown control directions based on fuzzy approximation. *IEEE Trans. Syst. Man Cybern. Syst.* 2016, 47, 1909–1918. [Crossref] [CrossRef]
- Rafael, D.; Andre, G. Nonlinear model predictive control applied to an autonomous underwater vehicle. *IEEE J. Ocean Eng.* 2020, 45, 799–812. [Crossref]
- Zhang, S.; Fan, L.T.; Gao, J.W.; Pu, J.; Xu, K. Fault diagnosis of underwater vehicle and design of intelligent self-rescue system. J. Coast. Res. 2018, 83, 872–875. [Crossref] [CrossRef]
- Yan, J.L.; Li, X.; Yang, X.Y.; Luo, X.; Hua, C.; Guan, X. Integrated localization and tracking for AUV with model uncertainties via scalable sampling-based reinforcement learning approach. *IEEE Trans. Syst. Man Cybern. Syst.* 2021, 52, 6952–6967. [Crossref] [CrossRef]
- 28. Zhu, D.; Wang, L.; Hu, Z.; Yang, S.X. A Grasshopper Optimization-based fault-tolerant control algorithm for a human occupied submarine with the multi-thruster system. *Ocean Eng.* **2021**, 242, 110101. [Crossref] [CrossRef]
- 29. Wang, X.M.; Zerr, B.; Thomas, H.; Clement, B.; Xie, Z. Pattern formation of multi-AUV systems with the optical sensor based on displacement-based formation control. *Int. J. Syst. Sci.* 2020, *51*, 348–367. [CrossRef]
- Ahmadi, K.; Asadi, D.; Merheb, A.; Nabavi-Chashmi, S.-Y.; Tutsoy, O. Active fault-tolerant control of quadrotor UAVs with nonlinear observer-based sliding mode control validated through hardware in the loop experiments. *Control. Eng. Pract.* 2023, 137, 105557. [Crossref] [CrossRef]
- 31. Yu, W.; Zhao, C. Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability. *IEEE Trans. Ind. Electron.* **2019**, *67*, 5081–5091. [Crossref] [CrossRef]
- Yu, W.; Zhao, C. Robust monitoring and fault isolation of nonlinear industrial processes using denoising autoencoder and elastic net. *IEEE Trans. Control. Syst. Technol.* 2019, 28, 1083–1091. [Crossref] [CrossRef]
- 33. Yu, W.; Zhao, C.; Huang, B. MoniNet with concurrent analytics of temporal and spatial information for fault detection in industrial processes. *IEEE Trans. Cybern.* **2021**, *52*, 8340–8351. [Crossref] [PubMed]

**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.