Fuzzy Fault-Tolerant Controller with Guaranteed Performance for MIMO Systems under Uncertain Initial State

ChunWu Yin, Saleem Riaz*, Ali Arshad Uppal, Jamshed Iqbal

Abstract: It is always problematic that the initial value of the trajectory tracking error must be inside the area included in the prescribed performance constraint function. To overcome this problem, a novel fault-tolerant control strategy is designed for a second-order Multi-Input and Multi-Output Nonlinear System (MIMO-NLS) with unknown initial states, actuator faults, and control saturation. Firstly, a predefined time convergence (PTC) stability criterion is theoretically proven. Then, an error conversion function is introduced to convert the trajectory tracking error to a new error variable with an initial value of zero, and an adaptive fuzzy system is designed to approximate the compound interference composed of actuator fault, parameter perturbation, control saturated overamplitude, and external disturbance. Based on the backstepping control method, prescribed performance control method, and predefined time convergence stability theory, an adaptive fuzzy fault-tolerant controller for the new error variable is designed and theoretically proven for the predefined time convergence of the closed-loop system. The numerical simulation results of the guaranteed performance trajectory tracking control for industrial robots with actuator faults demonstrate that the adaptive fuzzy fault-tolerant control algorithm has strong fault tolerance to actuator faults and anti-interference capabilities. The convergence time and performance of trajectory tracking errors can be preset in advance, and the parameter settings of the prescribed performance constraint function are not affected by the initial state values.

Keywords: Robot control, fault-tolerant control, predefined time convergence, prescribed performance control (PPC), fuzzy system.

1. INTRODUCTION

For uncertain nonlinear systems that are highly complex in nature, only designing a controller to maintain system stability can no longer meet practical engineering requirements. More and more systems have started to pay attention to the improvement of controllability and suppression of steady-state error, overshoot, convergence time, and passive fault tolerance. Thus, the prescribed performance control (PPC) method based on the prescribed performance function (PPF) is widely utilized in the guaranteed performance control of all nonlinear systems. However, the parameter setting of the PPF is related to the initial value of the system state. When the initial value of the system state is outside the areas included in the PPF, the guaranteed performance control strategy will fail, resulting in the system instability. Studying the guaranteed performance control strategy with unknown initial state value is of great theoretical significance and engineering application value.

The PPC method is widely utilized in the guaranteed performance control of different nonlinear systems because it can ensure that the overshoot of the trajectory tracking error is less than a set value. At the same time, it converges at no less than a certain set convergence speed and finally converges to a predetermined small range [1, 2]. In paper [3], for robot systems with control input saturation constraints and parameter uncertainties, a performance-preserving control strategy is designed by using deep neural networks to approximate the uncertain parts of the system. In paper [4], a finite time performance function with sine function is constructed, and fuzzy logic system is utilized for approximating the uncertain part of the second-order nonlinear system. Then, a guaranteed performance control algorithm with finite time convergence is designed for second-order nonlinear system. Paper [5] designed a prescribed performance controller with selectable transient performance for nonlinear systems by an error conversion method so that the output of the closed-loop system converges according to the trajectory of the preset curve. Although the existing guaranteed performance control strategy can make the controlled system have prescribed performance, its design premise is that the initial state value of all controlled

Manuscript received; revised; accepted. Recommended by Associate Editor under the direction of Editor. This journal was supported by the Korean Federation of Science and Technology Societies Grant.

ChunWu Yin is with the School of Information and Control Engineering, Xi'an University of Architecture and Technology, Shaanxi, Xi'an 710055, China (e-mail: yincwxa2013@mail.nwpu.edu.cn). Saleem Riaz is with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: saleemriaznwpu@mail.nwpu.edu.cn). Ali Arshad Uppal is with the Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad, Pakistan (e-mail: ali_arshad@comsats.edu.pk). Jamshed Iqbal is with the School of Computer Science, Faculty of Science and Engineering, University of Hull, UK (e-mail: j.iqbal@hull.ac.uk).
* Corresponding author: Saleem Riaz.

systems must be inside the areas included in the PPF. Estimation of initial state value is always challenging in numerous engineering applications and even this value will change greatly according to the actual application background. Setting a larger parameter value of performance constraint function will reduce the control effect of the controlled system and waste more control energy. Setting a smaller parameter value of performance constraint function will limit the application range of the controller. To solve the problem of uncertain initial state in the PPC, the current paper introduces an error conversion function to convert the arbitrary initial value of state to the origin, so that the initial value of the transformed variable is located at the included areas of PPF, and gives a performance-preserving control strategy to solve the problem of uncertain initial state.

The study introduces a cutting-edge and resilient strategy designed for global PDT tracking control of robot manipulators under inherent uncertainties [6]. Taking into consideration the effects of uncertainties and external interferences on trajectory tracking performance, the strategy employs a nonsingular robust control method, complemented by a nonlinear vector. This combination ensures that both positional and velocity tracking errors converge to the origin within a PDT. Key merits of this approach include its structural simplicity, straightforward implementation, and provision of an exact controller parameter to determine convergence time. The study also unveils a robust controller adept at ensuring precise trajectory adherence in robotic operations without requiring a temporal component within the trajectory definition [7]. By leveraging a robust velocity field and a controller uninfluenced by specific robot parameters, the proposed framework guarantees accurate trajectory tracking within a predefined time. Addressing the challenges of robust trajectory tracking for robot manipulators having uncertainties, the present study introduces a novel sliding surface and crafts a robust control mechanism to guarantee global approximate fixed-time convergence [8]. The analysis confirms that positional errors are bound to converge globally to a compact set around zero within a uniformly limited timeframe, subsequently decaying at an exponential rate. The proposed technique boosts faster transient responses, enhanced steady-state accuracy, and singularity-free operation, among other benefits. However, it utilized a fixed time sliding surface. In paper [9], the authors delineated a control strategy tailored for redundant manipulators, ensuring both PDT convergence of position purposes and robustness against model uncertainties and external disturbances. By harnessing dynamic consistency, this method realizes a rigid task hierarchy, empowering the robotic system to meet desired endpoints within its functional space, even amidst obstacles and joint constraints. Furthermore, article [10] presented a novel prescribed performance-tracking control mechanism for uncertain robotic manipulators, ensuring stability within a finite timeframe. It utilized a modified integral nonlinear SM plane and a super-twisting control law. In this study, scholars investigated that the system achieves precise trajectory tracking control. Notable features comprised estimated convergence rates, preset boundaries for maximum overshoot, and predetermined steady-state control error margins. Moreover, the controller optimized control torque without compromising robustness. The effectiveness of this control strategy was exemplified through industrial robot manipulator simulations, demonstrating global stability and finite-time convergence. Paper [11] presented a trajectory tracking controller for robotic manipulators, uniquely designed to operate without a dedicated model. It ensured tracking error convergence within a predefined interval, irrespective of the initial conditions. Utilizing the backstepping technique, the controller methodically determined the evolution law of the tracking error. The model perceived manipulator dynamics and disruptions as undetermined components, employing time-delay estimation for their identification. Exact mathematical validation confirmed the controller's global uniform prescribed-time stability. Its efficacy was further validated through simulations on a 2-DOF robot manipulator. Many scholars have explored rigorous hybrid methods employing predefined time control, such as complex neural control. These frameworks primarily focus on fixed and finite time [12, 13]. To address this research void, we have developed an innovative approach for a fuzzy fault-tolerant controller, ensuring optimal performance for MIMO-NLS.

Numerous control algorithms are utilized in the guaranteed performance control design of nonlinear systems. Traditional PID control has defects, including unsatisfactory suppression effect of time-varying uncertainties and low control accuracy. Scholars used adaptive technology to enhance classical PID, which effectively improved the robustness of nonlinear systems to uncertain factors. A PI fault-tolerant controller with guaranteed performance was designed for nonlinear systems by using adaptive law to estimate the control gains of PI controller in [14]. Scholars are also trying to propose robust control for linear time-varying systems with combination of multiagent technology [15, 16]. Furthermore, for backstepping control [17], sliding mode control [18], neural network control [19], fuzzy control [20, 21], and other nonlinear control methods combined with the PPC methods, different PPC controllers [22] are proposed successively to effectively enhance the trajectory tracking transient performance of nonlinear systems. However, these methods can only make the trajectory tracking errors converge to zero asymptotically. In practical control, it is essential to improve the convergence speed of nonlinear system. To enhance the trajectory tracking speed of nonlinear system, the finite time convergence control theory is applied to the trajectory tracking control of nonlinear system with prescribed performance. In paper [23], a finite-time convergence controller with guaranteed performance is designed for nonlinear systems with dead zones. In actual scenario, the tracking errors of the nonlinear systems must converge to a preset area within

the prescribed performance requirements. In paper [24], a finite time control strategy with guaranteed performance based on event-triggered method is designed for linear systems experiencing the external perturbations, and finite time convergence is achieved. Although the finite time control can accelerate the convergence rate of trajectory tracking, the convergence time depends on the initial value of the system and the control parameters, making the convergence time uncontrollable [25]. The fixed time control theory was proposed by scholars to improve the controllability of the system's convergence time [26, 27]. The upper bound of the convergence time is independent of the initial value of the system. This theory is combined with PPF to design a guaranteed performance controller such that the convergence time of the trajectory tracking error is only dependent on controller parameters. System stability is determined by the control parameters that also affect the convergence time. The convergence time of the system is still difficult to set in advance, and it is hard to meet the needs of some engineering applications that especially require predetermined convergence. Therefore, scholars put forward the convergence concept of predefined time. In the PTC theory, the time of trajectory tracking error is regardless of the system parameters and initial state, and the convergence time can be set arbitrarily to make the system convergent as well as controllable. The mentioned study and the convergence theory of predefined time are introduced in [28]. By introducing acceleration function, a Lyapunov stability condition for a predefined time with an upper bound of convergence time is independent of the control parameters, and the initial value of the system is given. In addition, a robust controller with PTC is designed so that the trajectory tracking error convergence time of nonlinear systems can be predetermined in advance.

Considering the above analysis, the current paper will design a fault-tolerant controller with predefined convergence time (PCT) and prescribed performance constraints of trajectory tracking errors for nonlinear systems with actuator faults, control saturation constraints, parameter perturbation, and external disturbance. The main innovations of this paper are as follows:

(1) The trajectory tracking error conversion function is introduced to convert the trajectory tracking error at any initial position into a new variable with an initial value at the origin so that the error changes according to the prescribed performance constraint function after a specified time, and the transient performance of trajectory tracking error is guaranteed. It solves the problem that the parameters of the PPF cannot be determined when the initial state value is unknown in the PPC.

(2) Using the theory of speed function, the Lyapunov stability condition for the PTC is given, and proof is also derived. The guaranteed trajectory tracking error convergence to any small region within the prescribed region is also given. Moreover, the theoretical basis of the convergence of predefined time is demonstrated.

(3) The control input with actuator fault and control saturation constraint was converted to unconstrained control input, and the adaptive fuzzy system was designed to approximate the compound interference composed of actuator fault, parameter perturbation, and control saturation over amplitude. The adaptive fuzzy fault-tolerant controller has been constructed by using the combination of the control theory of backstepping and the predefined time. The theory proves that the predefined time stability of the closed-loop system enhances the passive fault tolerance of the nonlinear systems to the actuator fault and the strong robustness to the uncertain factors.

The structure of the current paper is given as follows. Section 2 describes the control problem, which is the unconstrained transformation of the control input with actuator fault and control saturation constraints. Section 3 introduces the transformation method of the PPF and the proof of the stability criterion of the PTC. Section 4 illustrates the main design of adaptive fuzzy fault-tolerant controller with PTC and its stability. Numerical simulation is given in Section 5 that verifies the proposed control and mainly focuses on the dynamic performance under faults. The comparative simulation analysis is carried out under the conditions that the initial state value is outside the included area of the PPF. Then, the initial state value is inside the included area of that function for which the convergence time is different. Finally, the comparative results are discussed in detail with other adaptive controllers to verify the effectiveness and superiority of our proposed algorithm.

2. CONTROL PROBLEM DESCRIPTION

2.1. Nonlinear System Mode

The generalized mathematical model of a MIMO-NLS with actuator faults is taken as a problem. Consider the following nonlinear model:

$$\begin{cases} \dot{\boldsymbol{x}}_1(t) = \boldsymbol{x}_2(t) \\ \dot{\boldsymbol{x}}_2(t) = \boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}(t), t) + \boldsymbol{B}(t)\boldsymbol{u}^F(t) + \boldsymbol{d}(t) \\ \boldsymbol{y}(t) = \boldsymbol{x}_1(t) \end{cases} \tag{1}$$

where $\boldsymbol{x}_1(t), \boldsymbol{x}_2(t) \in R^m$ are the states of MIMO-NLS, denoting $\boldsymbol{x}(t) = [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)]^{\mathrm{T}}$; $\boldsymbol{\theta}(t)$ is the system parameter; $\boldsymbol{y}(t) \in \mathbf{R}^m$ represents the system output; $\boldsymbol{u}^F(t) \in \mathbf{R}^m$ is the control input of the system having actuator fault; $\boldsymbol{d}(t) \in \mathbf{R}^m$ symbolizes an external disturbance of the system; $\boldsymbol{B}(t)$ is taken as a bounded invertible function matrix having appropriate dimension. Furthermore, there exists a constant $M_1 > 0$ such that the following Lipschitz condition is satisfied:

$$\|\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}, t) - \boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{x}_{\mathrm{d}}, t)\| \le M_1 \| \boldsymbol{x} - \boldsymbol{x}_{\mathrm{d}} \| \tag{2}$$

2.2. Actuator Fault Model with Control Saturation Constraint

The control input of the actual system is generated by the actuator, and an association of the control input and the torque is produced by the actuator which can be described as follows:

$$\boldsymbol{u}^F(t) = \boldsymbol{H}\boldsymbol{u}_{\mathrm{d}}(t) \tag{3}$$

where $\boldsymbol{H} \in \mathbf{R}^{m\times n}$ is the installation matrix of the actuator; $\boldsymbol{u}_{\mathrm{d}}(t) \in \mathbf{R}^{n}$ is the vector formed by the control torque generated by an actuator, which is the control input actually entered by the actuator into the nonlinear system. In practical engineering applications, actuator faults will occur due to wear and aging of actuator, and output bounded constraints will exist due to physical structure limitations.

Common actuator faults comprise failure fault and additive deviation fault. Considering the influence of the actuator fault, the relationship between the actual control input $\boldsymbol{u}_{\mathrm{d}}(t)$ and the desired control input $\boldsymbol{u}_{\mathrm{c}}(t)$ of the actuator is as follows:

$$\begin{aligned}\boldsymbol{u}_{\mathrm{d}}(t) &= \boldsymbol{F}_a\boldsymbol{u}_{\mathrm{c}}(t)+\bar{\boldsymbol{u}}(t)\\ &= \boldsymbol{u}_{\mathrm{c}}(t)+\left((\boldsymbol{F}_a-I)\boldsymbol{u}_{\mathrm{c}}(t)+\bar{\boldsymbol{u}}(t)\right)\end{aligned} \tag{4}$$

In the above calculation, $\boldsymbol{u}_{\mathrm{c}}(t)$ depicts the control input from the actuator to the nonlinear system under the fault-free ideal state. $\bar{\boldsymbol{u}}(t)$ indicates the additive deviation fault of the actuator. $\boldsymbol{F}_a=\mathrm{diag}(f_{a1},f_{a2},\cdots,f_{an})$ is the failure loss factor matrix of the actuator. The element f_{ai} satisfies $0\le f_{ali}\le f_{ai}\le f_{aui}$, for which constants f_{ali} and f_{aui} are bounds, and $f_{aui}\ge 1$; $f_{ai}=0$ shows the complete failure of the actuator; $0<f_{ai}<1$ designates that the i th actuator is invalid; $f_{ai}=1$ specifies that the i th actuator works normally; $f_{ai}>1$ indicates that the i th actuator is stuck, resulting in the increase of the fault gain; $f_{ai}\ne 1$ depicts that the i th actuator is faulty. $\boldsymbol{I}$ is the identity matrix.

Note 1: In Formula (4), the control input $\boldsymbol{u}_{\mathrm{d}}(t)$ with failure fault and additive deviation fault is converted into the sum of the actual input $\boldsymbol{u}_{\mathrm{c}}(t)$ of the system and the fault part $\boldsymbol{\delta}(t)=(\boldsymbol{F}_a-I)\boldsymbol{u}_{\mathrm{c}}(t)+\bar{\boldsymbol{u}}(t)$.

Limited by the physical structure and safety requirements of the actuator, the control input $\boldsymbol{u}_{\mathrm{c}}(t)$ under the fault-free ideal state of the actuator has a bounded constraint; that is, the maximum value $u_{\max}$ exists, and each component $\boldsymbol{u}_{\mathrm{c}}(t)$ is satisfied:

$$u_{\mathrm{ci}}=\begin{cases}u_{\mathrm{ci}} & |u_{\mathrm{ci}}|\le u_{\max}\\ u_{\max}\mathrm{sign}(u_{\mathrm{ci}}) & |u_{\mathrm{ci}}|>u_{\max}\end{cases} \tag{5}$$

To eliminate the influence of the control input saturation constraints on the design of fault-tolerant controllers, unconstrained input $\boldsymbol{u}(t)$ is introduced, and the deviation between bounded constraint input $\boldsymbol{u}_{\mathrm{c}}(t)$ and unconstrained input $\boldsymbol{u}(t)$ is denoted as follows:

$$\Delta\boldsymbol{u}(t)=\boldsymbol{u}_{\mathrm{c}}(t)-\boldsymbol{u}(t) \tag{6}$$

By substituting equation (6) and equation (4) into equation (3), the actuator fault model with control saturation constraints is obtained:

$$\boldsymbol{u}^{F}(t)=\boldsymbol{H}\boldsymbol{u}(t)+\boldsymbol{H}\Delta\boldsymbol{u}(t)+\boldsymbol{H}\boldsymbol{\delta}(t) \tag{7}$$

When system parameter $\boldsymbol{\theta}(t)$ has parameter perturbation, let $\boldsymbol{\theta}(t)=\boldsymbol{\theta}_0(t)+\Delta\boldsymbol{\theta}(t)$, where $\boldsymbol{\theta}_0(t)$ denotes the nominal parameter and $\Delta\boldsymbol{\theta}(t)$ is the parametric perturbation part. Then, the parametric perturbation and actuator fault model are substituted into the second-order nonlinear system:

$$\begin{cases}\dot{\boldsymbol{x}}_1(t)=\boldsymbol{x}_2(t)\\ \dot{\boldsymbol{x}}_2(t)=\boldsymbol{f}(\boldsymbol{\theta}_0,\boldsymbol{x}(t),t)+\boldsymbol{B}(t)\boldsymbol{H}\boldsymbol{u}(t)+\boldsymbol{w}(t)\\ \boldsymbol{y}(t)=\boldsymbol{x}_1(t)\end{cases} \tag{8}$$

where $\boldsymbol{w}(t)=\boldsymbol{f}(\Delta\boldsymbol{\theta},t)+\boldsymbol{B}(t)\left(\boldsymbol{H}\Delta\boldsymbol{u}(t)+\boldsymbol{H}\boldsymbol{\delta}(t)\right)+\boldsymbol{d}(t)$ is the compound interference item comprised of parameter perturbation, actuator fault, control saturation constraint, and external disturbance.

The main objective is to design a fault-tolerant controller $\boldsymbol{u}(t)$ for the second-order nonlinear system (1) including parametric perturbation, external disturbance, and actuator fault so that the output $\boldsymbol{y}(t)$ of system (1) can precisely track the given signal $\boldsymbol{y}_{\mathrm{d}}(t)$ within any predefined time T_s.

3. PRESCRIBED PERFORMANCE CONSTRAINT FUNCTION TRANSFORMATION AND PREDEFINED TIME CONVERGENCE

3.1. Prescribed Performance Constraint Function Transformation

For the derivation of the tracking error and mathematical proof of the proposed controller, we can describe the following form of error which is the error between the trajectory tracking and measured output of the system.

$$\boldsymbol{e}(t)=\boldsymbol{y}(t)-\boldsymbol{y}_{\mathrm{d}}(t) \tag{9}$$

To ensure that the trajectory tracking error has good transient performance such as convergence rate, overshoot, and steady-state error in the whole control process, the PPF is generally adopted to limit the tracking error $\boldsymbol{e}(t)=[e_1(t),\cdots,e_m(t)]^{\mathrm{T}}$ within the boundary range of the PPF [29]; that is, the trajectory tracking error $e_i(t)$ satisfies

$$-\rho(t)<e_i(t)<\rho(t) \tag{10}$$

where the expression of the PPF $\rho(t)$ is

$$\rho(t)=(\rho_0-\rho_\infty)\mathrm{e}^{-k_\rho t}+\rho_\infty \tag{11}$$

where parameters $\rho_0,\rho_\infty,k_\rho$ denote the preset positive constants. ρ_0 denotes the initial value of the PPF; ρ_∞ is the steady-state value of that function, regulating the final steady-state interval of the performance; k_ρ determines the convergence rate of the PPF.

In the existing PPC strategy, the premise to ensure

the establishment of $-\rho(t) < e_i(t) < \rho(t)$ is that the initial value of tracking error $e_i(0)$ must meet $|e_i(0)| < \rho_0$. Otherwise, the controller will fail. For the guaranteed effectiveness of the controller, it is vital to approximate the initial amount $e_i(0)$ of the error to determine the value ρ_0 in the performance function during the controller design. However, it is difficult to estimate the initial value of the controlled system in some engineering applications, and even the initial value of some nonlinear systems will change randomly. Thus, it is impossible to reset the parameter ρ_0 in the performance function every time. To solve this problem, the tracking error $\boldsymbol{e}(t)$ is transformed as follows:

$$\boldsymbol{z}_1(t) = \varphi(t)\boldsymbol{e}(t) \tag{12}$$

where $\boldsymbol{z}_1(t)$ is the conversion error of tracking error $\boldsymbol{e}(t)$. The conversion function $\varphi(t)$ is defined as follows:

$$\varphi(t) = \begin{cases} \sin(\frac{\pi}{2}\frac{t}{T_\varphi}) & 0 \le t \le T_\varphi \\ 1 & t \ge T_\varphi \end{cases} \tag{13}$$

where T_φ is the parameter to be set in the conversion function and T_φ determines the latest time for error $\boldsymbol{e}(t)$ to enter the prescribed performance area. Obviously, the transformation function $\varphi(t)$ increases monotonically and satisfies $0 \le \varphi(t) \le 1$. According to the definition of $\varphi(t)$, $\boldsymbol{z}_1(0) = 0$ is always true regardless of the value of $\boldsymbol{e}(0)$, which makes the selection of parameter ρ_0 in the PPF more free. After time $t \ge T_\varphi$, $\varphi(t) = 1$, then $\boldsymbol{z}_1(t) = \boldsymbol{e}(t)$. If the fault tolerant controller can make $|z_{1i}(t)| < \rho(t)$ valid, then the tracking error $|e_i(t)| < \rho(t)$ is always valid after the predefined time $t = T_\varphi$.

After the introduction of transformation function $\varphi(t)$, the transformation error vector $\boldsymbol{z}_1(t)$ satisfies the performance function constraint in $t \ge 0$. However, the error $\boldsymbol{e}(t)$ satisfies the performance constraint only after $t \ge T_\varphi$. In practical engineering applications, when an initial cost of the error $\boldsymbol{e}(0)$ is completely unknown, it is not strictly required that the trajectory tracking error $\boldsymbol{e}(t)$ meets the performance constraint $|e_i(t)| < \rho(t)$ at $t = 0$. However, that trajectory error $\boldsymbol{e}(t)$ turns convergent according to the transient performance of the PPF after the predefined time T_φ. Therefore, transformation function $\varphi(t)$ is utilized for converting the error $\boldsymbol{e}(t)$ into a new conversion error vector $\boldsymbol{z}_1(t)$. It is feasible to design fault tolerant controller based on $\boldsymbol{z}_1(t)$.

3.2. Stability Condition for the Predefined Time Convergence

In this section, a function and conditions will be determined for the PTC of the tracking error. Such kind of function for PTC can be taken in the form of a speed function [30, 31] as follows:

$$\vartheta(t) = \begin{cases} \dfrac{T_s^{\,h}\kappa(t)}{(1-b_f)(T_s-t)^h + b_f T_s^{\,h}\kappa(t)}, & 0 \le t < T_s \\ \dfrac{1}{b_f}, & t \ge T_s \end{cases} \tag{14}$$

where the constants $0 < b_f < 1$, $h > 2$; $T_s > 0$ is the default convergence time; $\kappa(t)$ is a separate increasing function with an initial value of $\kappa(0) = 1$. Generally, $\kappa(t) = 1 + t^2$ or $\kappa(t) = e^t$ can be selected. The velocity function $\vartheta(t)$ has the following properties:

1） $\vartheta(t)$ increases monotonically in $t \in [0, T_s]$ and does not change in $t \in [T_s, \infty)$. Then, $\vartheta(t)$ is bounded; that is, $1 \le \vartheta(t) \le 1/b_f$.

2） $\vartheta(t)$ has a continuous and bounded derivative $\vartheta^{(i)}(t)$ up to order $h-1-i$, $(i = 0,1,2,\cdots,h-1)$.

3） Let $\tau(t) = \vartheta^{-1}(t)$. Then, $\tau(t)$ has a continuous and bounded derivative $\tau^{(i)}(t)$ up to order $h-2-i$, $(i = 0,1,2,\cdots,h-2)$.

Taking the derivative of equation (14), we obtain

$$\dot{\vartheta}(t) = \begin{cases} \dfrac{T_s^{\,h}(1-b_f)(T_s-t)^{h-1}[(T_s-t)\dot{\kappa}(t) + h\kappa(t)]}{[(1-b_f)(T_s-t)^h + b_f T_s^{\,h}\kappa(t)]^2}, & 0 \le t < T_s \\ 0, & t \ge T_s \end{cases} \tag{15}$$

***Lemma* 1 [32]:** Consider a first-order system $\dot{\boldsymbol{\eta}} = \boldsymbol{h}(t, \boldsymbol{\eta})$, where $\boldsymbol{\eta} = (\boldsymbol{z}, \boldsymbol{v})^{\mathrm{T}}$, function $\boldsymbol{h}(t, \boldsymbol{\eta})$ is piecewise continuous with respect to t and satisfies the local Lipschitz condition with respect to $\boldsymbol{\eta}$. For any positive number b_z, it is assumed that there are continuously differentiable positive definite functions $U(\boldsymbol{v})$ and $V_1(\boldsymbol{z})$ such that $V_1(\boldsymbol{z}) \to \infty$ for $\|\boldsymbol{z}\| \to b_z$ and $\gamma_1(\|\boldsymbol{v}\|) \le U(\boldsymbol{v}) \le \gamma_2(\|\boldsymbol{v}\|)$ hold (where $\|\boldsymbol{z}\|$ is the norm of L_2 space and $\gamma_1(\boldsymbol{v}), \gamma_2(\boldsymbol{v})$ are a class K_∞ function). Let $V(\boldsymbol{\eta}) = V_1(\boldsymbol{z}) + U(\boldsymbol{v})$, and $\|\boldsymbol{z}(0)\| < b_z$. If the inequality

$$\dot{V}(\boldsymbol{\eta})=\frac{\partial V}{\partial \boldsymbol{\eta}}\boldsymbol{h}\le -\lambda V+\Delta \tag{16}$$

holds，where the paramaters $\lambda,\Delta>0$, then, there is $\|\boldsymbol{z}(t)\|<b_z$ for $t\in[0,\infty)$, and $\boldsymbol{v}$ is bounded.

***Lemma* 2 [32]:** For arbitrary positive number b_z and arbitrary vector $\boldsymbol{z}(t)$ satisfying $\|\boldsymbol{z}(t)\|<b_z$, the following inequality is always true:

$$\log\left(\frac{b_z^2}{b_z^2-\boldsymbol{z}^{\mathrm{T}}(t)\boldsymbol{z}(t)}\right)\le\frac{\boldsymbol{z}^{\mathrm{T}}(t)\boldsymbol{z}(t)}{b_z^2-\boldsymbol{z}^{\mathrm{T}}(t)\boldsymbol{z}(t)} \tag{17}$$

To make it easier to design a controller with a PTC behavior for nonlinear systems, a theorem has been established, presenting Lyapunov stability conditions for convergence. This criterion is provided in the form of a theorem and has been backed up by theoretical proof.

***Theorem 1*:** For arbitrary predefined time $T_s>0$ and arbitrary infinitesimal constant $\varepsilon>0$, if the non-negative continuous differentiable function $V(t)$ is satisfied

$$\dot{V}(t)\le-(\beta_1+\beta_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})V(t)+C \tag{18}$$

where the parameters $\beta_1>1,\beta_2>1,C>0$, function $\vartheta(t)$ is defined as equation (14). If b_f,β_2 is chosen appropriately, $V(t)\le\varepsilon$ will always exist at time $t\ge T_s$.

Proof: According to equation (18),

$$\begin{aligned}&\dot{V}(t)\le-(\beta_1+\beta_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})V(t)+C\\&\Rightarrow\frac{\mathrm{d}(V(t)-C_1)}{\mathrm{d}t}+\beta_2\frac{\dot{\vartheta}(t)}{\vartheta(t)}(V(t)-C_1)\le-\beta_1(V(t)-C_1)\\&\Rightarrow\frac{\mathrm{d}(\vartheta(t)^{\beta_2}(V(t)-C_1))}{\mathrm{d}t}\le-\beta_1\vartheta(t)^{\beta_2}(V(t)-C_1)\\&\Rightarrow V(t)\le C_1+\vartheta(t)^{-\beta_2}e^{-\beta_1 t}(V(0)-C_1)\end{aligned} \tag{19}$$

where $C_1=C\Big/\left(\beta_1+\beta_2\frac{\dot{\vartheta}(t)}{\vartheta(t)}\right)$; if $t\ge T_{s2}$, $\dot{\vartheta}(t)=0$ holds, then $C_1=C/\beta_1$; if the value of β_1 is large enough, C_1 can approach any small positive number. According to equation (14), we can get $\lim\limits_{t\to T_s}\vartheta^{-1}(t)=\lim\limits_{t\to T_s}\frac{(1-b_f)(T_s-t)^h+b_f T_s^{\,h}\kappa(t)}{T_s^{\,h}\kappa(t)}=b_f$. Then, $\lim\limits_{t\to T_s}\vartheta^{-\beta_2}(t)=b_f^{\beta_2}$ holds. For the reason of $0<b_f,\beta_2>1$, if the value of b_f is small enough, and the value of β_2 is large enough, then, the value of $b_f^{\beta_2}$ is going to be very small. Because $e^{-\beta_1 t}<1$ and $V(0)$ is bounded, if $0<b_f<1,\beta_2>1$, $\beta_1>1$ are chosen appropriately, function $V(t)$ approaches any infinitesimal number ε at time $t\to T_s$, meaning $\lim\limits_{t\to T_s}V(t)=\varepsilon$. Since $V(t)>0$ and $\dot{V}(t)<0$, meaning $V(t)$ decreases monotonically, combined with $\lim\limits_{t\to T_s}V(t)=\varepsilon$, we know that when $t>T_s$, we also have $V(t)\le\varepsilon$.

4. DESIGN OF ADAPTIVE FUZZY FAULT-TOLERANT CONTROLLER

Fig. 1 depicts the overall control diagram of the proposed adaptive fuzzy fault-tolerant controller with PTC characteristics. Adaptive fuzzy fault-tolerant controller is composed of equivalent controller and adaptive fuzzy compensation controller. The equivalent controller is designed by the PPC method to ensure the PTC and the preset transient performance of the trajectory tracking error of the MIMO nominal system. In the adaptive fuzzy system, the weighted vector is updated according to the system state, and the compensator is generated to compensate the compound interference items $\boldsymbol{w}(t)$ of the system.

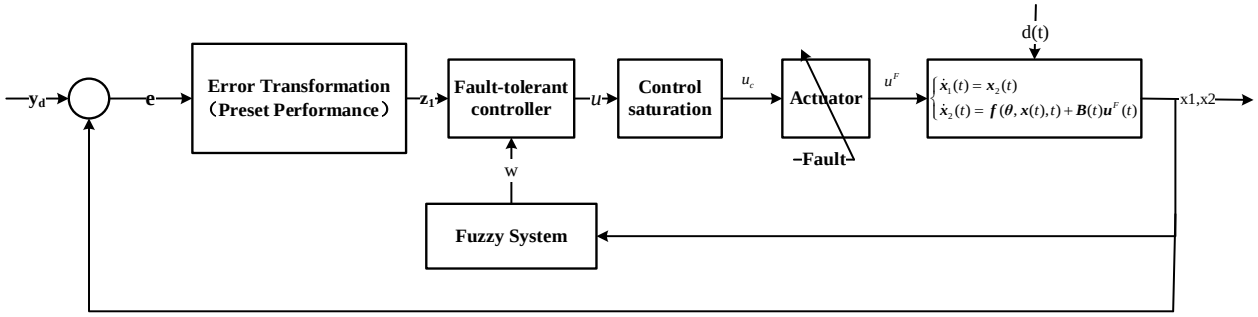


Fig. 1. Structure diagram of adaptive fuzzy fault-tolerant control.

4.1 Design of Equivalent Fault-Tolerant Controller with Predefined Time Convergence

The first and the second time derivative of $\boldsymbol{z}_1(t)=\varphi(t)\boldsymbol{e}(t)$ are

$$\dot{\boldsymbol{z}}_1(t)=\dot{\varphi}(t)\boldsymbol{e}(t)+\varphi(t)(\boldsymbol{x}_2(t)-\dot{\boldsymbol{y}}_{\mathrm{d}}(t)) \tag{20}$$

$$\begin{aligned}\ddot{\boldsymbol{z}}_1(t)&=\ddot{\varphi}(t)\boldsymbol{e}(t)+2\dot{\varphi}(t)(\boldsymbol{x}_2(t)-\dot{\boldsymbol{y}}_{\mathrm{d}}(t))\\&\quad+\varphi(t)(\dot{\boldsymbol{x}}_2(t)-\ddot{\boldsymbol{y}}_{\mathrm{d}}(t))\\&=\ddot{\varphi}(t)\boldsymbol{e}(t)+2\dot{\varphi}(t)(\boldsymbol{x}_2(t)-\dot{\boldsymbol{y}}_{\mathrm{d}}(t))\\&\quad+\varphi(t)(\boldsymbol{f}(\boldsymbol{\theta}_0,\boldsymbol{x}(t),t)+\boldsymbol{B}(t)\boldsymbol{H}\boldsymbol{u}(t)\\&\quad+\boldsymbol{w}(t)-\ddot{\boldsymbol{y}}_{\mathrm{d}}(t))\end{aligned} \tag{21}$$

The backstepping control method is utilized for designing an equivalent fault-tolerant controller with PTC. Firstly, the virtual controller $\boldsymbol{\alpha}(t)$ of $\boldsymbol{x}_2(t)$ is designed for subsystem $\dot{\boldsymbol{x}}_1(t)=\boldsymbol{x}_2(t)$.

Let $\boldsymbol{z}_2(t)=\boldsymbol{x}_2(t)-\boldsymbol{\alpha}(t)$. Consider Lyapunov function $V_1(t)=\frac{1}{2}\log\frac{\rho^2(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}$ and its time derivative

$$\begin{aligned}\dot{V}_1(t)&=\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)(\dot{\varphi}(t)\boldsymbol{e}(t)+\varphi(t)(\boldsymbol{x}_2(t)-\dot{\boldsymbol{y}}_{\mathrm{d}}(t))}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}-\frac{\dot{\rho}(t)\boldsymbol{z}_1(t)}{\rho(t)})\\&=\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\varphi(t)\boldsymbol{x}_2(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}(\dot{\varphi}(t)\boldsymbol{e}(t)\\&-\varphi(t)\dot{\boldsymbol{y}}_{\mathrm{d}}(t)-\frac{\dot{\rho}(t)\boldsymbol{z}_1(t)}{\rho(t)})\\&\le\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\varphi(t)\boldsymbol{x}_2(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{1}{2\lambda}\\&+\left(\frac{(\lambda b_{\varphi}^2\boldsymbol{e}^{\mathrm{T}}(t)\boldsymbol{e}(t)+\lambda\varphi^2(t)b_{\dot{\boldsymbol{y}}_{\mathrm{d}}}^2}{(\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t))}-\frac{\dot{\rho}(t)}{\rho(t)}\right)\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}\end{aligned}\tag{22}$$

where constant b_{φ} is the bound of $\dot{\varphi}(t)$, constant $b_{\dot{\boldsymbol{y}}_{\mathrm{d}}}$ is the bound of , and parameter λ is an arbitrary positive number. According to the mathematical expression of the performance function $\rho(t)$, inequality $-\dot{\rho}(t)>0$ holds，then $\dot{\boldsymbol{y}}_{\mathrm{d}}(t)$ $\varpi=\frac{(\lambda b_{\dot{\boldsymbol{y}}_{\mathrm{d}}}^2\boldsymbol{e}^{\mathrm{T}}(t)\boldsymbol{e}(t)+\lambda\varphi^2(t)b_{\dot{\boldsymbol{y}}_{\mathrm{d}}}^2}{(\rho^2(t)-\boldsymbol{z}^{\mathrm{T}}(t)\boldsymbol{z}(t))}-\frac{\dot{\rho}(t)}{\rho(t)}>0$. Since $b_{\dot{\varphi}}$ and $b_{\dot{\boldsymbol{y}}_{\mathrm{d}}}$ are unknown, so ϖ is unknown. To make the controller have stronger self-adaptability, $\hat{\varpi}$ is set as the estimate of ϖ, and the estimation error is denoted as $\tilde{\varpi}=\varpi-\varphi(t)\hat{\varpi}$. A new Lyapunov function $V_2(t)=V_1(t)+0.5r^{-1}\tilde{\varpi}^2$ is constructed. Then,

$$\begin{aligned}\dot{V}_2(t)\le&\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\varphi(t)\boldsymbol{x}_2(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{\varpi\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}\\&+\frac{1}{2\lambda}-\frac{1}{2r}\tilde{\varpi}(\dot{\varphi}(t)\hat{\varpi}+\varphi(t)\dot{\hat{\varpi}})\end{aligned}\tag{23}$$

The virtual controller of $\boldsymbol{x}_2(t)$ is given as

$$\begin{cases}\boldsymbol{\alpha}(t)=-(k_1+k_2\dfrac{\dot{\vartheta}(t)}{\vartheta(t)}+\hat{\varpi})\boldsymbol{z}_1(t)\\\dot{\hat{\varpi}}=-2\left(\bar{k}_1+\bar{k}_2\dfrac{\dot{\vartheta}(t)}{\vartheta(t)}+\dot{\varphi}(t)\right)\hat{\varpi}+\dfrac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)r}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}\end{cases}\tag{24}$$

where k_1,k_2 indicate the parameters of the virtual controller to be designed and $\bar{k}_1=\varphi(t)k_1,\bar{k}_2=\varphi(t)k_2$. If the value of $\boldsymbol{x}_2(t)$ is $\boldsymbol{\alpha}(t)$, the virtual controller (24) is substituted into equation (23), and then

$$\begin{aligned}\dot{V}_2(t)\le&\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\varphi(t)\left(-(k_1+k_2\dfrac{\dot{\vartheta}(t)}{\vartheta(t)}+\hat{\varpi})\boldsymbol{z}_1(t)\right)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}\\&+\frac{\varpi\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{1}{2\lambda}-\frac{1}{r}\tilde{\varpi}(\dot{\varphi}(t)\hat{\varpi}+\varphi(t)\dot{\hat{\varpi}})\\&=-(\varphi(t)k_1+\varphi(t)k_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}\\&+\frac{-\varphi(t)\hat{\varpi}\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t))}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{\varpi\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{1}{2\lambda}\\&-\frac{1}{r}\tilde{\varpi}(\dot{\varphi}(t)\hat{\varpi}+\varphi(t)\dot{\hat{\varpi}})\\&\le-(\bar{k}_1+\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{1}{2\lambda}\\&+\frac{(\varpi-\varphi(t)\hat{\varpi})\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t))}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}-\frac{1}{r}\tilde{\varpi}(\dot{\varphi}(t)\hat{\varpi}+\varphi(t)\dot{\hat{\varpi}})\\&=-(\bar{k}_1+\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{1}{2\lambda}\\&+\frac{\tilde{\varpi}\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t))}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}-\frac{1}{r}\tilde{\varpi}(\dot{\varphi}(t)\hat{\varpi}+\varphi(t)\dot{\hat{\varpi}})\end{aligned}$$

$$\begin{aligned}\dot{V}_2(t)\le&-(\bar{k}_1+\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{1}{2\lambda}\\&+\tilde{\varpi}\left(\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t))}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}-\frac{\dot{\varphi}(t)\hat{\varpi}+\varphi(t)\dot{\hat{\varpi}}}{r}\right)\\&=-(\bar{k}_1+\bar{k}_2\frac{\dot{\mu}(t)}{\mu(t)})\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{1}{2\lambda}+\\&\tilde{\varpi}\left(\begin{array}{l}\dfrac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t))}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\dfrac{\varphi(t)}{r}\left(\dot{\varphi}(t)+2\bar{k}_1+2\bar{k}_2\dfrac{\dot{\vartheta}(t)}{\vartheta(t)}\right)\hat{\varpi}\\-\dfrac{\dot{\varphi}(t)\hat{\varpi}}{r}-\dfrac{\varphi(t)\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}\end{array}\right)\\&=-(\bar{k}_1+\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}+\frac{1}{2\lambda}+\\&\tilde{\varpi}\left(\begin{array}{l}\dfrac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t))(1-\varphi(t))}{\rho^2(t)-\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}-\dfrac{\dot{\varphi}(t)(1-\varphi(t))\hat{\varpi}}{r}\\+\dfrac{\varphi(t)\hat{\varpi}}{r}\left(2\bar{k}_1+2\bar{k}_2\dfrac{\dot{\vartheta}(t)}{\vartheta(t)}\right)\end{array}\right)\end{aligned}\tag{25}$$

We can observe that when time t exceeds the time T_{φ}, and whenever a tracking error arrives the included area of the prescribed function, at least there is $1-\varphi(t)=0$. In addition, according to

$$\tilde{\varpi}\hat{\varpi}=\tilde{\varpi}(\varpi-\tilde{\varpi})\le-\frac{1}{2}\tilde{\varpi}^2+\frac{1}{2}\varpi^2\tag{26}$$

if $t\ge T_{\varphi}$, there is

$$\dot{V}_2(t) \le -(\bar{k}_1 + \bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t) - \boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}$$
$$+ \frac{\varphi(t)\tilde{\varpi}\hat{\varpi}}{r}(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) + \frac{1}{2\lambda}$$
$$\le -(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{1}{2} \log \frac{\boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}{\rho^2(t) - \boldsymbol{z}_1^{\mathrm{T}}(t)\boldsymbol{z}_1(t)}$$
$$-(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\mu}(t)}{\mu(t)}) \frac{\tilde{\varpi}^2}{2r} + (2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r} + \frac{1}{2\lambda}$$
$$\le -(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) V_2(t) + (2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r} + \frac{1}{2\lambda} \quad (27)$$

Since $(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r} + \frac{1}{2\lambda}$ is bounded, and it is a positive number, according to ***Theorem 1***, if the controller parameter is properly selected, $V_2(t)$ will always have $V_2(t) \le \varepsilon$ at time $t \ge T_s \ge T_\varphi$, where ε denotes an arbitrary small constant, which means that $\boldsymbol{z}_1(t)$ has a convergent nature and approaches to zero within the predefined time T_s. Combined with the properties of the barrier Lyapunov function described in ***Lemma 1***, we can observe that $\boldsymbol{z}_1(t)$ meets the performance function constraint $|z_{1i}(t)| < \rho(t)$. Combined with the definition of $\boldsymbol{z}_1(t)$, the tracking error $\boldsymbol{e}(t)$ satisfies the performance constraint $|e_i(t)| < \rho(t)$ at $t \ge T_\varphi$.

Then, the fault-tolerant controller is designed for subsystem $\dot{\boldsymbol{x}}_2(t) = \boldsymbol{f}(\theta_0, t) + \boldsymbol{B}(t)\boldsymbol{H}\boldsymbol{u}(t) + \boldsymbol{w}(t)$. Denote $\boldsymbol{z}_2(t) = \boldsymbol{x}_2(t) - \boldsymbol{\alpha}(t)$, and consider Lyapunov function as $V_3(t) = V_2(t) + \frac{1}{2}\mathrm{z}_2^{\mathrm{T}}(t)\mathrm{z}_2$, and we can obtain:

$$\dot{V}_3(t) = \dot{V}_2(t) + \boldsymbol{z}_2^{\mathrm{T}}(t)(\dot{\boldsymbol{x}}_2(t) - \dot{\boldsymbol{\alpha}}(t))$$
$$\le -(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) V_2(t) + (2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r} + \frac{1}{2\lambda}$$
$$+ \boldsymbol{z}_2^{\mathrm{T}}(t)(\boldsymbol{f}(\theta_0, \boldsymbol{x}(t), t) + \boldsymbol{B}(t)\boldsymbol{H}\boldsymbol{u}(t) + \boldsymbol{w}(t) - \dot{\boldsymbol{\alpha}}(t)) \quad (28)$$

The proposed controller is given as

$$\boldsymbol{u}(t) = \boldsymbol{H}^{-1}\boldsymbol{B}^{-1}(t) \begin{pmatrix} -(\bar{k}_1 + \bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \boldsymbol{z}_2(t) \\ -\boldsymbol{f}(\theta_0, \boldsymbol{x}(t), t) + \dot{\boldsymbol{\alpha}}(t) - \hat{\boldsymbol{w}}(t) \end{pmatrix} \quad (29)$$

Where $\boldsymbol{H}^{-1}$ is the generalized inverse of $\boldsymbol{H}$. $\hat{\boldsymbol{w}}(t)$ is the estimation of $\boldsymbol{w}(t)$. Substituting controller (29) into equation (28), we get

$$\dot{V}_3(t) \le -(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) V_2(t) + (2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r}$$
$$+ \frac{1}{2\lambda} + \boldsymbol{z}_2^{\mathrm{T}}(t)(\boldsymbol{f}(\theta_0, \boldsymbol{x}(t), t) + \boldsymbol{B}(t)\boldsymbol{H}\boldsymbol{u}(t) + \boldsymbol{w}(t) - \dot{\boldsymbol{\alpha}}(t))$$
$$= -(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) V_2(t) + (2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r} + \frac{1}{2\lambda}$$
$$-(\bar{k}_1 + \bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \boldsymbol{z}_2^{\mathrm{T}}(t)\boldsymbol{z}_2(t) + \boldsymbol{z}_2^{\mathrm{T}}(t)\tilde{\boldsymbol{w}}(t)$$
$$= -(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) V_2(t) + (2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r} + \frac{1}{2\lambda}$$
$$-(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{1}{2} \boldsymbol{z}_2^{\mathrm{T}}(t)\boldsymbol{z}_2(t) + \boldsymbol{z}_2^{\mathrm{T}}(t)\tilde{\boldsymbol{w}}(t)$$
$$= -(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) V_3(t) + (2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r}$$
$$+ \frac{1}{2\lambda} + \boldsymbol{z}_2^{\mathrm{T}}(t)\tilde{\boldsymbol{w}}(t)$$

where $\tilde{\boldsymbol{w}}(t)$ is the estimated error of compound interference; that is, $\tilde{\boldsymbol{w}}(t) = \boldsymbol{w}(t) - \hat{\boldsymbol{w}}(t)$. If the fuzzy system can accurately approximate $\boldsymbol{w}(t)$, there is $\tilde{\boldsymbol{w}}(t) = 0$. Since $(2\bar{k}_1 + 2\bar{k}_2 \frac{\dot{\vartheta}(t)}{\vartheta(t)}) \frac{\varpi^2}{2r} + \frac{1}{2\lambda}$ is positive and bounded, according to **Theorem 1**, if the controller parameters are properly selected, $V_3(t)$ always has $V_3(t) \le \varepsilon_2$ at time $t > T_s \ge T_\varphi$, where ε_2 is any small constant. Obviously, $\boldsymbol{z}_2(t)$ converges to any small area of origin at $t > T_s$; that is, $\boldsymbol{x}_2(t)$ also converges to that area having virtual control input $\boldsymbol{\alpha}(t)$ within a predefined time T_s. According to the property of barrier Lyapunov function, $\boldsymbol{z}_1(t)$ satisfies the performance function constraint $|z_{1i}(t)| < \rho(t)$, and the tracking error $\boldsymbol{e}(t)$ also satisfies the performance constraint $|e_i(t)| < \rho(t)$ at $t \ge T_\varphi$.

4.2 Adaptive Fuzzy System Approximates Compound Interference

The compound interference $\boldsymbol{w}(t)$ is approximated by a fuzzy system composed of single value fuzzification, product inference machine, and barycenter average defuzzification. Compound interference is the expression of $\boldsymbol{x}_1, \boldsymbol{x}_2$. Therefore, the input of the fuzzy system is $\boldsymbol{a} = [\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}]^{\mathrm{T}} \in R^{2m}$. It is sumed that the fuzzy system is composed of N fuzzy rules. The i^{th} fuzzy rule R^{i} is as follows: If a_1 is μ_1^i, …… , and a_{2m} is μ_{2m}^i, then S is b^i ($i = 1, 2, \cdots, N$). As given in the above equation, μ_j^i represents a membership function of

a_j $(j=1,2,\cdots,m)$; then, the output of fuzzy system can be given as follows:

$$s=\frac{\sum_{i=1}^{N}\zeta_i\prod_{j=1}^{2m}\mu_j^i(a_j)}{\sum_{i=1}^{N}\prod_{j=1}^{2m}\mu_j^i(a_j)}=\boldsymbol{\chi}^{\mathrm{T}}(\boldsymbol{a})\boldsymbol{\zeta} \tag{30}$$

where $\chi_i(\boldsymbol{a})=\prod_{j=1}^{2m}\mu_j^i(a_j)\Big/\sum_{i=1}^{N}\prod_{j=1}^{2m}\mu_j^i(a_j)$, $\boldsymbol{\chi}=[\chi_1(\boldsymbol{a}),\chi_2(\boldsymbol{a}),\cdots,\chi_N(\boldsymbol{a})]^{\mathrm{T}}$, $\boldsymbol{\zeta}=[\zeta_1,\zeta_2,\cdots,\zeta_N]^{\mathrm{T}}$. When the fuzzy system is utilized to approach the compound interference $\boldsymbol{w}(t)$, the fuzzy system is designed in the form of respective approaching $w_1(t),w_2(t),\cdots,w_m(t)$; that is, the fuzzy system is utilized to estimate the components of the compound interference $\boldsymbol{w}(t)$.

$$\hat{w}_1(t)=\frac{\sum_{i=1}^{N}\zeta_{1i}\prod_{j=1}^{2m}\mu_{1j}^i(a_j)}{\sum_{i=1}^{N}\prod_{j=1}^{2m}\mu_{1j}^i(a_j)}=\boldsymbol{\chi}_1{}^{\mathrm{T}}(\boldsymbol{a})\boldsymbol{\zeta}_1$$

$$\hat{w}_2(t)=\frac{\sum_{i=1}^{N}\zeta_{2i}\prod_{j=1}^{2m}\mu_{2j}^i(a_j)}{\sum_{i=1}^{N}\prod_{j=1}^{2m}\mu_{2j}^i(a_j)}=\boldsymbol{\chi}_2{}^{\mathrm{T}}(\boldsymbol{a})\boldsymbol{\zeta}_2$$

$$\hat{w}_k(t)=\frac{\sum_{i=1}^{N}\zeta_{ki}\prod_{j=1}^{2m}\mu_{kj}^i(a_j)}{\sum_{i=1}^{N}\prod_{j=1}^{2m}\mu_{kj}^i(a_j)}=\boldsymbol{\chi}_k{}^{\mathrm{T}}(\boldsymbol{a})\boldsymbol{\zeta}_k$$

which can be re-written as

$$\hat{\boldsymbol{w}}=\begin{bmatrix}\hat{w}_1\\ \hat{w}_2\\ \vdots\\ \hat{w}_m\end{bmatrix}=\begin{bmatrix}\boldsymbol{\chi}_2{}^{\mathrm{T}}(\boldsymbol{a}) & 0 & \cdots & 0\\ 0 & \boldsymbol{\chi}_2{}^{\mathrm{T}}(\boldsymbol{a}) & \cdots & 0\\ \vdots & \cdots & \ddots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{\chi}_2{}^{\mathrm{T}}(\boldsymbol{a})\end{bmatrix}\begin{bmatrix}\zeta_1\\ \zeta_2\\ \vdots\\ \zeta_m\end{bmatrix}=\boldsymbol{\Phi}(\boldsymbol{a})\boldsymbol{\zeta} \tag{31}$$

When the optimal weighted vector $\boldsymbol{\zeta}^*$ of the fuzzy system is obtained, the fuzzy system can accurately approximate the compound interference item $\boldsymbol{w}(t)$; that is,

$$\boldsymbol{w}=\boldsymbol{\Phi}(\boldsymbol{a})\boldsymbol{\zeta}^* \tag{32}$$

In the actual process, it is impossible to obtain the optimal weighting vector $\boldsymbol{\zeta}^*$ and only its estimate $\hat{\boldsymbol{\zeta}}$ can be obtained. Thus, the adaptive modification law of the weighting vector of the fuzzy system is

$$\dot{\hat{\boldsymbol{\zeta}}}=\eta\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{a})\boldsymbol{z}_2-\eta(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\hat{\boldsymbol{\zeta}} \tag{33}$$

where $\eta>0$ is the parameter to be designed.

Note 1: In many studies, weight vectors of fuzzy approximation system do not converge or converge asymptotically. However, the weight vector designed in this paper can converge in a predefined time.

4.3 Stability Analysis

Theorem 2: Considering MIMO-NLS (1) with uncertain parameters, external disturbance, actuator fault, and control saturation constraints, if the adaptive fuzzy systems (31) and (33) approach compound interference, the prescribed performance fault-tolerant controller (29) based on adaptive fuzzy systems can ensure that the trajectory tracking errors of the nonlinear system are uniformly bounded within the predefined time, and the trajectory tracking error has preset transient and steady-state performance.

Proof: Denoting the weight error of the fuzzy system as $\tilde{\boldsymbol{\zeta}}=\boldsymbol{\zeta}^*-\hat{\boldsymbol{\zeta}}$, construct a Lyapunov function as

$$V_4(t)=V_3(t)+\frac{1}{2\eta}\tilde{\boldsymbol{\zeta}}^{\mathrm{T}}\tilde{\boldsymbol{\zeta}} \tag{34}$$

The time derivative of the Lyapunov function is

$$\begin{aligned}
\dot{V}_4(t)&=\dot{V}_3(t)-\eta^{-1}\tilde{\boldsymbol{\zeta}}^{\mathrm{T}}\dot{\hat{\boldsymbol{\zeta}}}\\
&\le-(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})V_3(t)+(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\varpi^2}{2r}\\
&+\frac{1}{2\lambda}+\boldsymbol{z}_2{}^{\mathrm{T}}(t)\tilde{\boldsymbol{w}}(t)-\frac{1}{\eta}\tilde{\boldsymbol{\zeta}}^{\mathrm{T}}\dot{\hat{\boldsymbol{\zeta}}}\\
&=-(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})V_3(t)+(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\varpi^2}{2r}\\
&+\frac{1}{2\lambda}+\tilde{\boldsymbol{\zeta}}^{\mathrm{T}}(\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{a})\boldsymbol{z}_2-\frac{\dot{\hat{\boldsymbol{\zeta}}}}{\eta})\\
&=-(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})V_3(t)+(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\varpi^2}{2r}\\
&+\frac{1}{2\lambda}+(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\tilde{\boldsymbol{\zeta}}^{\mathrm{T}}\hat{\boldsymbol{\zeta}}\\
&\le-(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})V_3(t)+(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})\frac{\varpi^2}{2r}\\
&+\frac{1}{2\lambda}+(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})(\frac{1}{2}\boldsymbol{\zeta}^{*\mathrm{T}}\boldsymbol{\zeta}^*-\frac{1}{2}\tilde{\boldsymbol{\zeta}}_{\zeta}^{\mathrm{T}}\tilde{\boldsymbol{\zeta}})\\
&\le-(2\bar{k}_1+2\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})V_4(t)+\Delta
\end{aligned} \tag{35}$$

where $\Delta=(\bar{k}_1+\bar{k}_2\frac{\dot{\vartheta}(t)}{\vartheta(t)})(\frac{\varpi^2}{r}+\boldsymbol{\zeta}^{*\mathrm{T}}\boldsymbol{\zeta}^*)+\frac{1}{2\lambda}>0$.

According to **Theorem 1**, if the controller parameters are properly selected, $V_4(t)$ always has $V_4(t)\le\varepsilon_3$ at time $t>T_s\ge T_\varphi$. In the above expression, ε_3 indicates a small constant. It depicts that the tracking error of MIMO-NLS is going to converge within the predefined time T_s. It is obvious that the tracking error $\boldsymbol{e}(t)$ also satisfies and meets the performance criterion such as $|e_i(t)|<\rho(t)$ at $t\ge T_\varphi$.

5. NUMERICAL SIMULATION ANALYSIS

Consider the dynamic equation of a 2DF manipulator with actuator fault, external disturbance,

and parameter perturbation [33, 34]:

$$\boldsymbol{D}(\boldsymbol{q}(t))\ddot{\boldsymbol{q}}(t)+\boldsymbol{C}(\boldsymbol{q}(t),\dot{\boldsymbol{q}}(t))\dot{\boldsymbol{q}}(t)+\boldsymbol{g}(\boldsymbol{q}(t))=\boldsymbol{u}^{F}(t)+\boldsymbol{d} \quad (36)$$

where $\boldsymbol{q}(t),\dot{\boldsymbol{q}}(t),\ddot{\boldsymbol{q}}(t)$ denote the angle, angular velocity, and angular acceleration of the manipulator, respectively. $\boldsymbol{u}^{F}(t)$ represents the input variable with actuator fault constraint and control input saturation constraint. The expression for inertia matrix $\boldsymbol{D}(\boldsymbol{q}(t))$, centripetal force matrix $\boldsymbol{C}(\boldsymbol{q}(t),\dot{\boldsymbol{q}}(t))$, gravity vector $\boldsymbol{g}(\boldsymbol{q}(t))$, and external disturbance $\boldsymbol{d}(t)$ are characterized as

$$\boldsymbol{D}(t)=\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, \boldsymbol{C}(t)=\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & 0 \end{bmatrix}, \boldsymbol{g}(t)=\begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

$$D_{11}=m_1 l_{c1}^2+m_2(l_1^2+l_{c2}^2+2l_1 l_{c2}\cos(q_2(t)))+I_1+I_2$$

$$D_{12}=m_2(l_{c2}^2+l_1 l_{c2}\cos(q_2(t))+I_2), D_{22}=m_2 l_{c2}^2+I_2$$

$$D_{21}=m_2(l_{c2}^2+l_1 l_{c2}\cos(q_1(t)+q_2(t)))$$

$$C_{11}=-m_2 l_1 l_2 \sin(q_2(t))q_2(t)$$

$$C_{12}=-m_2 l_1 l_2 \sin(q_2(t))[q_1(t)+q_2(t)]$$

$$C_{21}=m_2 l_1 l_2 \sin(q_2(t))q_1(t)$$

$$g_1=(m_1 l_{c1}+m_2 l_1)g\cos(q_1(t))+m_2 l_{c2} g\cos(q_1(t)+q_2(t)), g_2=m_2 l_{c2} g\cos(q_1(t)+q_2(t))$$

$$\boldsymbol{d}(t)=0.2\begin{bmatrix} 3\cos(t)+4\sin(0.3t) \\ -1.5\sin(0.2t)+3\cos(0.5t) \end{bmatrix}$$

In the overall design of robot, the nominal system of robot arm is often utilized to design the controller of robot arm. The inertia matrix, centripetal force, and gravity matrix are decomposed into the sum of nominal matrix and perturbation momentum as follows: $\boldsymbol{D}(t)=\boldsymbol{D}_0(t)+\Delta\boldsymbol{D}(t)$, $\boldsymbol{C}(t)=\boldsymbol{C}_0(t)+\Delta\boldsymbol{C}(t)$, $\boldsymbol{g}(t)=\boldsymbol{g}_0(t)+\Delta\boldsymbol{g}(t)$, where $\boldsymbol{D}_0(t)$, $\boldsymbol{C}_0(t)$, $\boldsymbol{g}_0(t)$ are nominal matrices and $\Delta\boldsymbol{D}(t)$, $\Delta\boldsymbol{C}(t)$, $\Delta\boldsymbol{g}(t)$ are perturbation matrices given as: $\Delta\boldsymbol{D}(t)=0.05\boldsymbol{D}_0, \Delta\boldsymbol{C}(t)=0.05\boldsymbol{C}_0, \Delta\boldsymbol{g}(t)=0.05\boldsymbol{g}_0$.

Suppose the maximum allowable input of the actuator is $\boldsymbol{u}_{\text{Max}}=300N.m$, and the actuator installation matrix $\boldsymbol{H}$, failure coefficient matrix $\boldsymbol{F}_a$, and deviation fault $\bar{\boldsymbol{u}}(t)$, respectively, are

$$\boldsymbol{H}=\begin{bmatrix} 1 & 1/3 \\ 0 & 1/3 \end{bmatrix}, \quad \boldsymbol{F}_a=\begin{bmatrix} 0.7 & 0 \\ 0 & 0.9 \end{bmatrix}, \quad \bar{\boldsymbol{u}}(t)=\begin{bmatrix} -0.2 \\ 0.1 \end{bmatrix} \text{N m}$$

The actuator fault and parameter perturbation were substituted into the manipulator model, and the nominal system model of the 2-degree-of-freedom manipulator was obtained as follows:

$$\boldsymbol{D}_0(t)\ddot{\boldsymbol{q}}(t)+\boldsymbol{C}_0(t)\dot{\boldsymbol{q}}(t)+\boldsymbol{g}_0(t)=\boldsymbol{u}(t)+\boldsymbol{w}(t) \quad (37)$$

From the above calculation, the expression $\boldsymbol{w}(t)=-\Delta\boldsymbol{D}(t)\ddot{\boldsymbol{q}}(t)-\Delta\boldsymbol{C}(t)\dot{\boldsymbol{q}}(t)-\Delta\boldsymbol{g}(t)+\boldsymbol{d}(t)+\Delta\boldsymbol{u}(t)+\boldsymbol{\eta}(t)$ denotes a compound interference. Let's take $\boldsymbol{x}_1(t)=\boldsymbol{q}(t)$, $\boldsymbol{x}_2(t)=\dot{\boldsymbol{q}}(t)$. The dynamic equation of the manipulator (37) is converted to

$$\begin{cases} \dot{\boldsymbol{x}}_1(t)=\boldsymbol{x}_2(t) \\ \dot{\boldsymbol{x}}_2(t)=-\boldsymbol{D}_0^{-1}(t)\boldsymbol{C}_0(t)\boldsymbol{x}_2(t)-\boldsymbol{D}_0^{-1}(t)\boldsymbol{g}_0(t) \\ \qquad +\boldsymbol{D}_0^{-1}(t)\boldsymbol{H}\boldsymbol{u}(t)+\boldsymbol{D}_0^{-1}(t)\boldsymbol{w}(t) \\ \boldsymbol{y}(t)=\boldsymbol{x}_1(t) \end{cases}$$

visible, $\boldsymbol{f}(\theta_0,\boldsymbol{x}(t),t)=-\boldsymbol{D}_0^{-1}(t)(\boldsymbol{C}_0(t)\boldsymbol{x}_2(t)+\boldsymbol{g}_0(t))$, $\boldsymbol{B}(t)=\boldsymbol{D}_0^{-1}(t)$. The membership degree functions of the fuzzy system are listed as follows:

$$\mu_1^i(\boldsymbol{a})=\mathrm{e}^{-(\boldsymbol{a}+\frac{\pi}{6})/(\frac{\pi}{24})^2}, \mu_2^i(\boldsymbol{a})=\mathrm{e}^{-(\boldsymbol{a}+\frac{\pi}{12})/(\frac{\pi}{24})^2}, \mu_3^i(\boldsymbol{a})=\mathrm{e}^{-\boldsymbol{a}/(\frac{\pi}{24})^2},$$

$$\mu_4^i(\boldsymbol{a})=\mathrm{e}^{-(\boldsymbol{a}-\frac{\pi}{12})/(\frac{\pi}{24})^2}, \mu_5^i(\boldsymbol{a})=\mathrm{e}^{-(\boldsymbol{a}-\frac{\pi}{6})/(\frac{\pi}{24})^2}$$

The input of the fuzzy system is $\boldsymbol{a}=[\boldsymbol{x}_1;\boldsymbol{x}_2]$, and there are 5^4 rules for fuzzy system. The relevant parameters in the nominal matrix of the manipulator are shown in Table 1.

Table 1. Parametric symbols and values of the robotic manipulator.

Parameter	Value	Unit
Robot arm's mass (m_1)	10	kg
Robot arm's mass (m_2)	5	kg
Robot arm's length of joint (l_1)	1.00	m
Robot arm's length of joint (l_2)	0.50	m
Length of joint (l_{c1})	0.50	m
Length of joint (l_{c2})	0.25	m
Moment of inertia (I_1)	0.83	kg.m^2
Moment of inertia (I_2)	0.30	kg.m^2
Gravitational acceleration (g)	9.81	m·s^{-2}

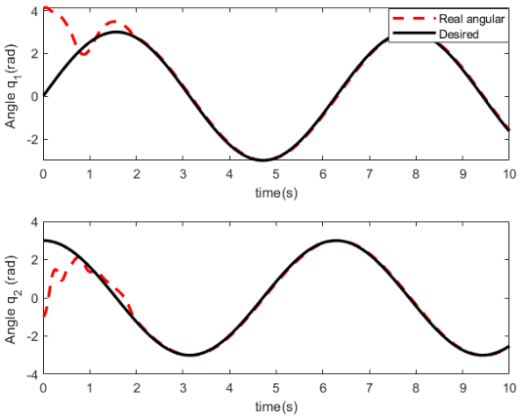
The parametric values used for the simulations of the adaptive fuzzy controller are given in Table 2.

Table 2. Demonstration of the controller parametric values.

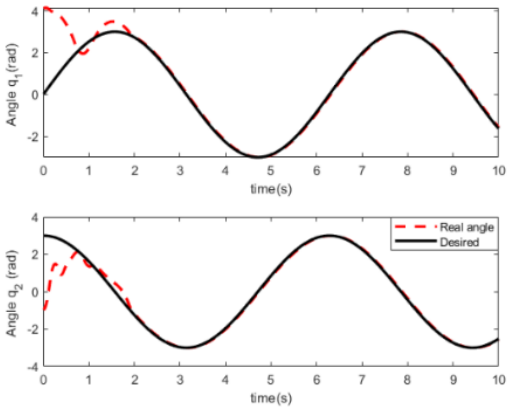
Element	Description
Total simulation time	10 s
Performance function	ρ_0=3; ρ_∞ =0.1; k_ρ=0.9
Transfer function	$T_{\varphi=2}$
Speed Function	h=5, κ(t)=1+t^2,b_f=0.008, T_s=3
Fault-tolerant controller	η=100.5;k_1=40;k_2=0.5; r=0.02
Reference position signal	y_{1d} =3 sin(3t) y_{2d} =3 cos(3t)

5.1 Simulation of Initial State Values Outside Prescribed Performance Constraint Functions

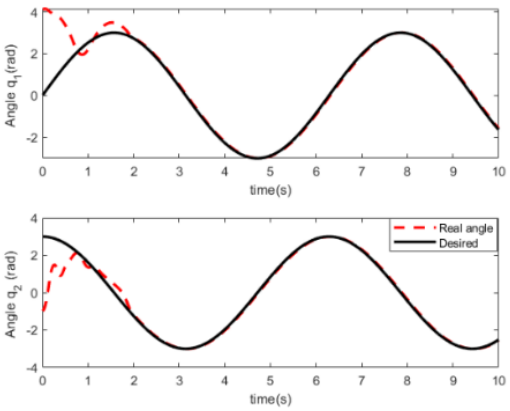
Let the initial angle and initial angular velocity of the manipulator be $q(0)=[4.1,-1]^{\mathrm{T}}, \dot{q}(0)=[1.5,1.5]^{\mathrm{T}}$, respectively. Obviously, the initial angle of the manipulator is outside the inclusion region of the performance constraint function. The following figures set different convergence rates kρ of performance functions for numerical simulation analysis. The simulation results are depicted in the following figures (see Fig. 2 to Fig. 6) having the parameters kρ=0.6, kρ=0.9, kρ=1.9 kρ=3.1.



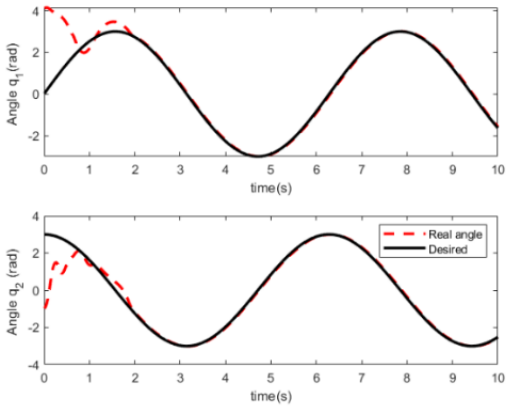
(a) k_ρ=0.6



(b) k_ρ=0.9

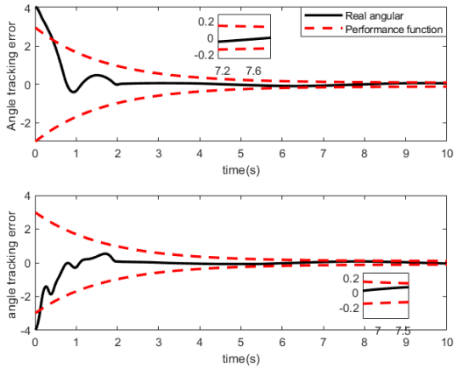


(c) k_ρ=1.9

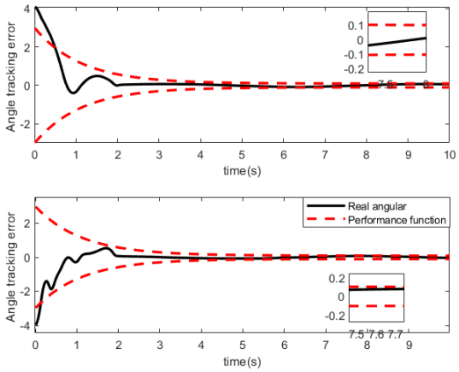


(d) k_ρ=3.1

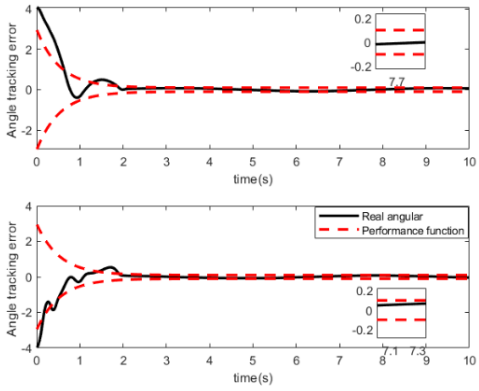
Fig. 2. Angle tracking curve under different convergence rates.



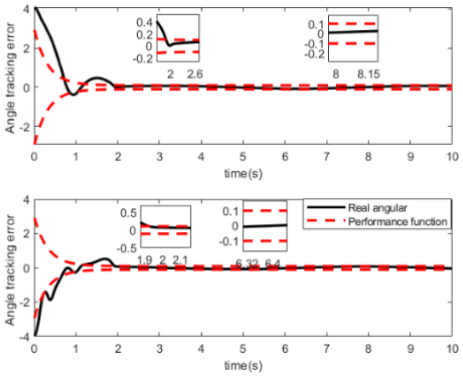
(a) k_ρ=0.6



(b) k_ρ=0.9



(c) k_ρ=1.9



(d) k_ρ=3.1

Fig. 3. Angle tracking error under different convergence rates.

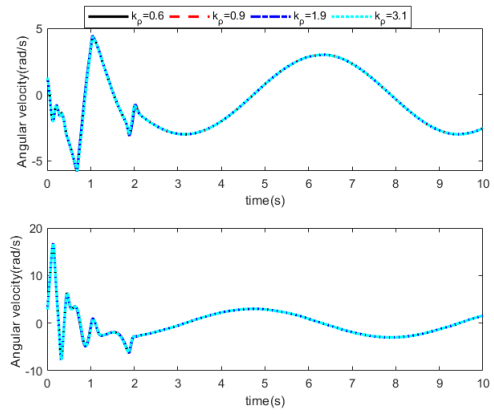


Fig. 4. Angular velocity tracking curve under different convergence rates.

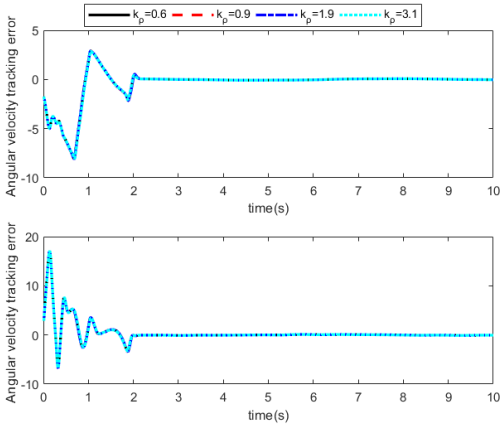


Fig. 5. Angular velocity tracking error under different convergence rates.

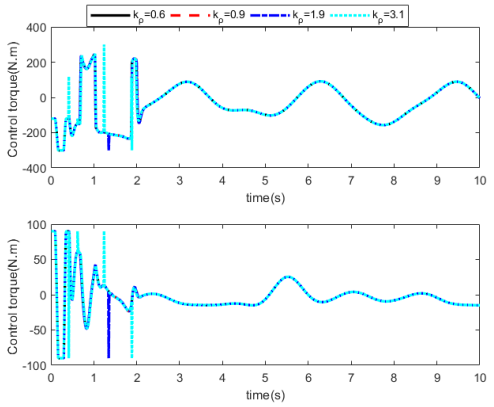


Fig. 6. Actual control input under different convergence rates.

Both Fig. 2 and Fig. 3 illustrate that angle tracking error of the manipulator gradually enters the inclusion area of that function from the outside, stays within the inclusion area of the PPF after the predefined time $T_\varphi = 2s$, and finally reaches the steady state within the $T_s = 3s$. A large convergence rate will cause angle tracking error to jump out of the included region of the PPF. However, it guarantees that angle tracking error remains within the included region of the PPF after the predefined time $T_\varphi = 2s$ and remains stable within that constraint function region during the preset convergence time $T_s = 3s$. Fig. 2 and Fig. 3 indicate that changing the convergence rate of the performance function has little impact on the convergence of the angular tracking error of the manipulator. This demonstrates that our proposed control has an advantage to solve the control problem if the initial state of the MIMO-NLS is outside the included region of the PPF.

Fig. 4 and Fig. 5 display that under the different convergence rates of performance functions, the velocity tracking errors of the manipulator can all converge and stabilize to 0 after 2.1s, and the convergence time is less than the PCT $T_s = 3s$. Simultaneously, changing the convergence speed of the PPF does not disturb the convergence characteristics of the angular velocity.

Fig. 6 shows that the value of the control input is less than the maximum value $\boldsymbol{u}_{\text{Max}} = 300N.m$ of the control saturation constraint, and there is no oscillation phenomenon. The variation of the convergence rate affects the input control torque. A larger convergence rate will result in a larger jump in the control torque before angle tracking error is convergent. When angle tracking error reaches a stable state, the trend of control torque under different convergence speeds is similar.

5.2 Simulation of Initial State Value in Prescribed Performance Constraint Function

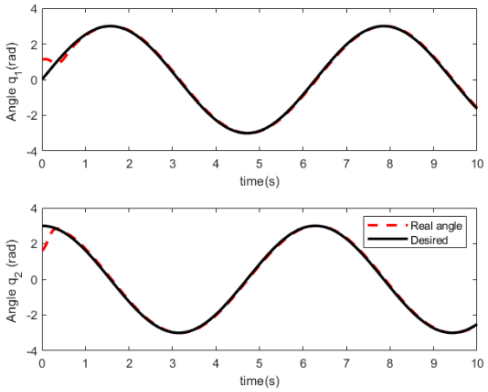
Suppose that the initial angle and initial angular velocity of the manipulator are $q(0) = [1.1,1.6]^{\mathrm{T}}, \dot{q}(0) = [1.5,1.5]^{\mathrm{T}}$, respectively. Obviously, the initial value angle is located inside the inclusion region of the PPF. The convergence speed of the performance constraint function is set as kρ=0.6, kρ=0.9, kρ=1.9, kρ=3.1, and the simulation results are shown in Fig. 7 to Fig. 11.

Fig. 7 and Fig. 8 demonstrate that angle tracking error of the manipulator is exponentially converged and reaches the steady state within 1s. The convergence time is less than T_s, and the angular velocity tracking error has a small overshoot, and there is no chattering. The actual simulation results demonstrate that, under the condition of the same control parameters, whenever an initial value of error is less than the initial value ρ_0 of the PPF, based on the fault-tolerant controller designed in this paper, then this error can converge to a stable state, and the initial value angle tracking error is less than the initial value ρ_0 of the PPF. The faster the convergence time of the tracking error of the angular velocity, the better the dynamic performance becomes.

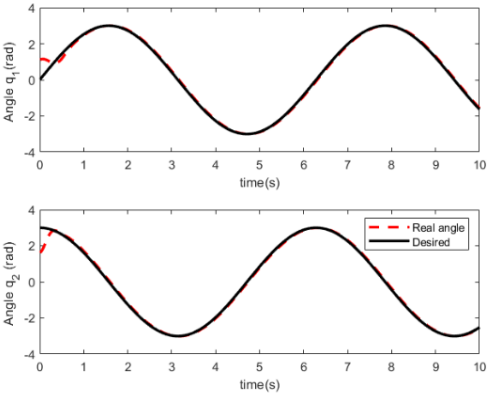
Fig. 9 and Fig. 10 illustrate that the convergence trajectory curve of angular velocity tracking error is relatively smooth without chattering. The convergence time of the tracking error of the angular velocity is 1 second, which is shorter than the specified convergence time of 3 seconds ($T_s = 3s$). Different convergence rates of PPFs have a little influence on the convergence trend and dynamic characteristics of angular velocity tracking errors.

Fig. 11 indicates that when the initial angle tracking

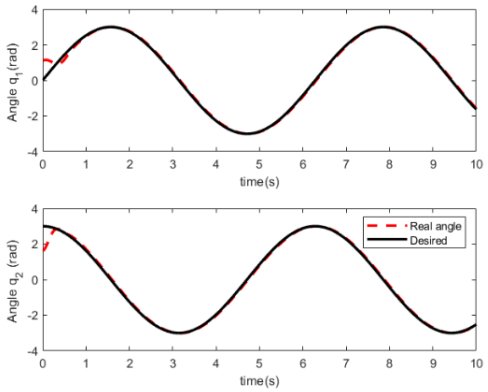
error of the manipulator is within the included region of the PPF, it takes less time for the actual control torque input to the manipulator to reach saturation. During the whole control process, the control input torque changes smoothly, and there is no chattering phenomenon. Changing the convergence speed of the performance function does not affect the variation trend of the control torque.



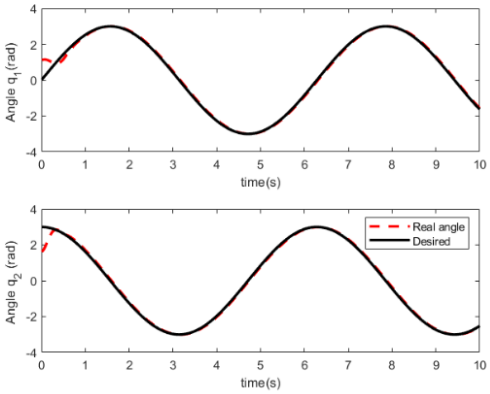
(a) k_ρ=0.6



(b) k_ρ=0.9

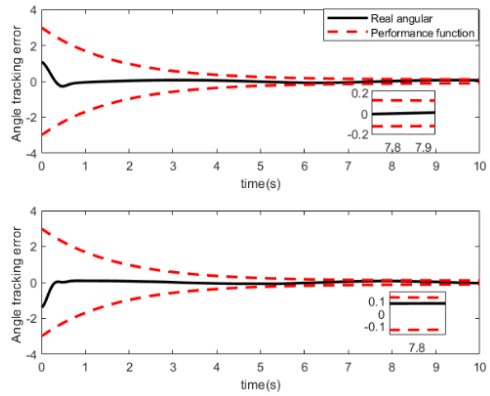


(c) k_ρ=1.9

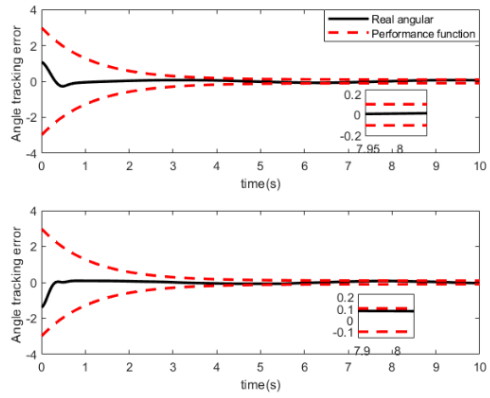


(d) k_ρ=3.1

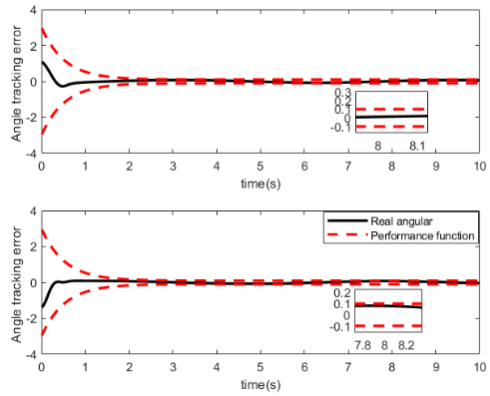
Fig. 7. Angle tracking curve under different convergence rates.



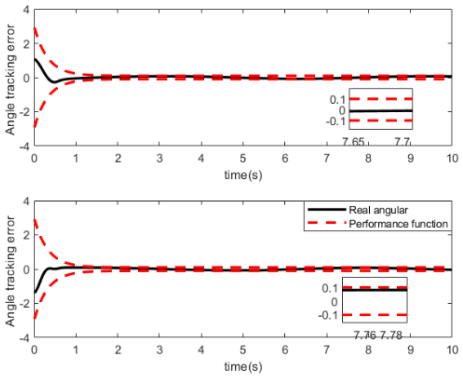
(a) k_ρ=0.6



(b) k_ρ=0.9



(c) k_ρ=1.9



(d) k_ρ=3.1

Fig. 8. Angle tracking error under different convergence rates.

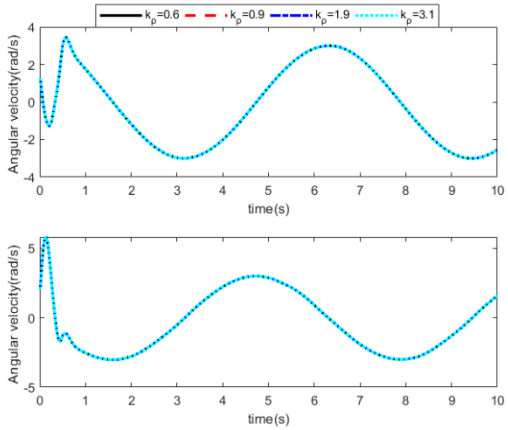


Fig. 9. Angular velocity tracking curve under different convergence rates.

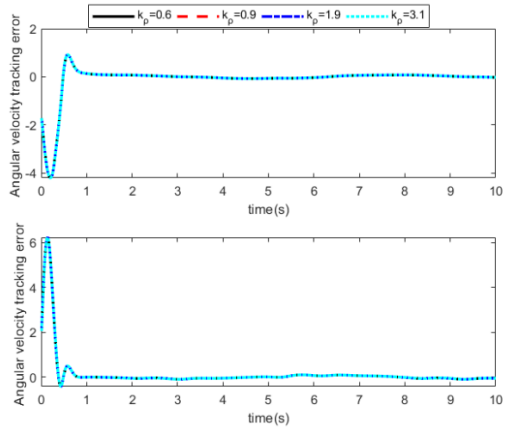


Fig. 10. Angular velocity tracking error under different convergence rates.

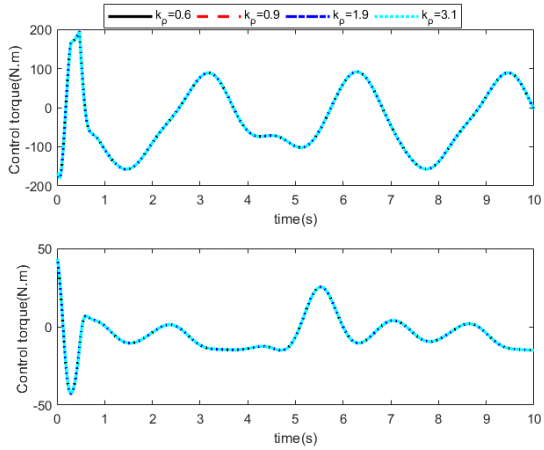


Fig. 11. Actual control input under different convergence rates.

5.3 Simulation of Different PCT

Let the initial values of the angle and angular velocities be set as $q(0)=[1.1,1.6]^{\mathrm{T}}, \dot{q}(0)=[1.5,1.5]^{\mathrm{T}}$, respectively. The convergence speed of performance function is set as k_ρ=0.9, the time T_φ in the conversion function $\varphi(t)$ is set to 1s, and the PCT T_s is set as 1.5s，2s，3s，and 4s correspondingly. The simulation results have been depicted in Fig. 12 to Fig. 16.

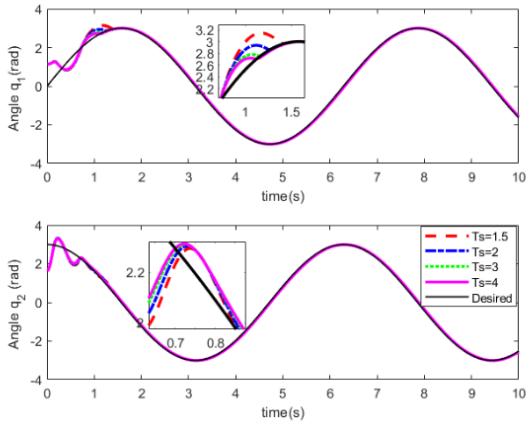


Fig. 12. Angle tracking curve under different predefined times.

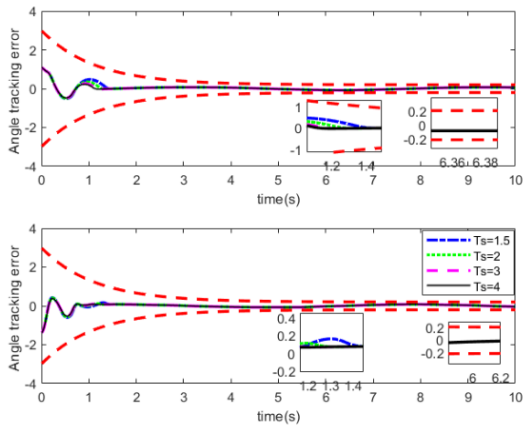


Fig. 13. Angle tracking error under different PCT.

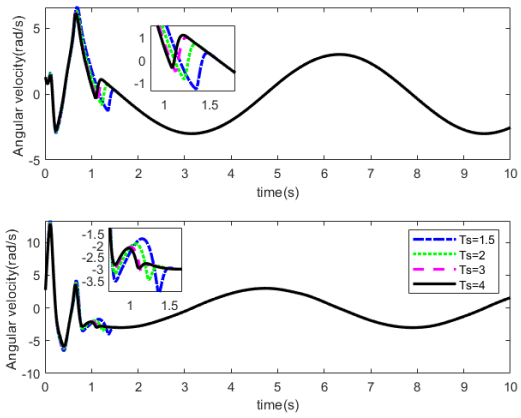


Fig. 14. Angular velocity tracking curve under different PCT.

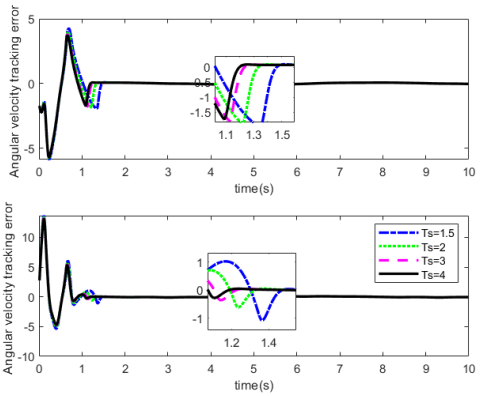


Fig. 15. Angular velocity tracking error under different PCT.

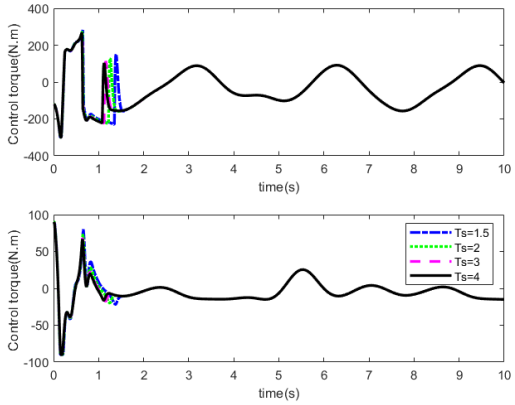


Fig. 16. Actual control input under different PCT.

Fig. 12 and Fig. 13 illustrate that the angle tracking error of the manipulator can still converge to a stable state before the PCT. Moreover, the convergence time of angle tracking errors is less than 1.5s, and the convergence curve is relatively smooth. The larger the convergence time, the faster the convergence speed angle tracking errors. Upon reaching a state of equilibrium, the simulation results demonstrate that the angle tracking errors remain constant despite any alteration to the controller's parameters and the PDCT $T_s < T_\varphi$. The robotic manipulator's angle tracking error gradually converges within the boundaries of the PPF and settles into a state of stability.

Fig. 14 and Fig. 15 reveal that the angular velocity tracking error of the manipulator reaches stability within 1.5 seconds following three adjustments with varying predetermined convergence times. This indicates that the fault-tolerant controller effectively enables the angular velocity of the manipulator to converge to zero within the PDCT. Furthermore, the results demonstrate that different predetermined convergence times do not significantly affect the convergence characteristics of angular velocity tracking errors, with a longer convergence time resulting in faster convergence of the angular velocity tracking errors.

Fig. 16 displays the variation trend of the actual control input torque of the manipulator under different PCT. In the whole control process, the control torque saturates initially but settles in a legitimate range as the state vector settles. When the angle tracking error of the manipulator does not reach a stable state, different PCT will affect the change of the control input torque. The smaller the PCT, the longer the change time of the control torque. When angle tracking error reaches a stable state, the control input torque under different PCT is the same.

5.4 Comparative Simulation of Different Control Methods

In order to highlight the effectiveness and advantages of the proposed algorithm, the controller in paper [17] is improved, and an adaptive law is designed to estimate complex interference items to compensate the influence of uncertain parts on the system. A prescribed performance sliding mode adaptive controller (PPSMAC) is designed as follows:

$$\begin{cases} \boldsymbol{u} = H^{-1}D_0\boldsymbol{M}_3^{-1}(-k_3\boldsymbol{S} - \hat{\boldsymbol{w}} - \boldsymbol{M}_1 - \boldsymbol{M}_2 - \boldsymbol{M}_3 f(x) + \boldsymbol{M}_3\ddot{y}_d - c\dot{\boldsymbol{\varepsilon}}) \\ \dot{\hat{\boldsymbol{w}}}(t) = k_4\boldsymbol{S}(t), \quad \boldsymbol{\varepsilon} = \frac{1}{2}\ln\frac{\rho(t)+\boldsymbol{e}(t)}{\rho(t)-\boldsymbol{e}(t)}, \quad \boldsymbol{S} = \dot{\boldsymbol{\varepsilon}} + c\boldsymbol{\varepsilon} \\ \boldsymbol{M}_1 = \mathrm{diag}(\frac{\ddot{\rho}(\rho+e_1)-(\dot{\rho}+\dot{e}_1)^2}{2(\rho+e_1)^2}, \frac{\ddot{\rho}(\rho+e_2)-(\dot{\rho}+\dot{e}_2)^2}{2(\rho+e_2)^2}) \\ \boldsymbol{M}_2 = \mathrm{diag}(-\frac{\ddot{\rho}(\rho-e_1)-(\dot{\rho}-\dot{e}_1)^2}{2(\rho-e_1)^2}, -\frac{\ddot{\rho}(\rho-e_2)-(\dot{\rho}-\dot{e}_2)^2}{2(\rho-e_2)^2}) \\ \boldsymbol{M}_3 = \mathrm{diag}(\frac{\rho+e_1}{2(\rho+e_1)^2}+\frac{\rho-e_1}{2(\rho-e_1)^2}, \frac{\rho+e_2}{2(\rho+e_2)^2}+\frac{\rho-e_2}{2(\rho-e_2)^2}) \end{cases} \tag{38}$$

where parameters $k_3, k_4 > 0$ are controller gains, and the trajectory tracking error is $\boldsymbol{e}(t) = [e_1(t), e_2(t)]^{\mathrm{T}}$.

The following simulation results compare the proposed algorithm with the prescribed performance sliding mode adaptive controller (PPSMAC). PTFTC control parameters are unchanged, and the parameters of performance function, the actuator fault, and the maximum value of control saturation constraint are the same as the relevant parameter settings in the PTFTC. The controller parameters of PPSMAC are selected as k3= 50; c=10; k4= 20.

The simulation results indicate that when the initial angle of the manipulator falls outside the boundaries of the PPF, the manipulator, utilizing the PPSMAC, is unable to converge. This indicates the limitations of the PPSMC in resolving control problems involving initial angles that are situated outside the defined performance function. When the initial angle of the manipulator is set within the PPF, the initial angle and angular velocity are established as $q(0) = [1.1, 1.6]^{\mathrm{T}}$, $\dot{q}(0) = [1.5, 1.5]^{\mathrm{T}}$.

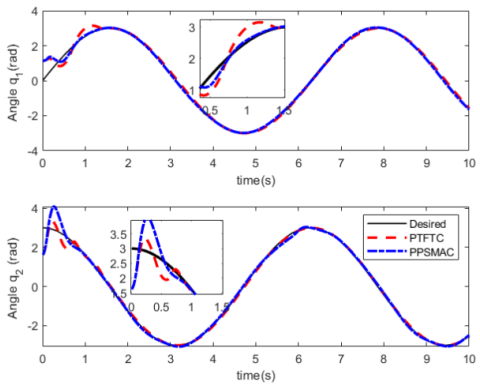


Fig. 17. Angle tracking curve under different controllers.

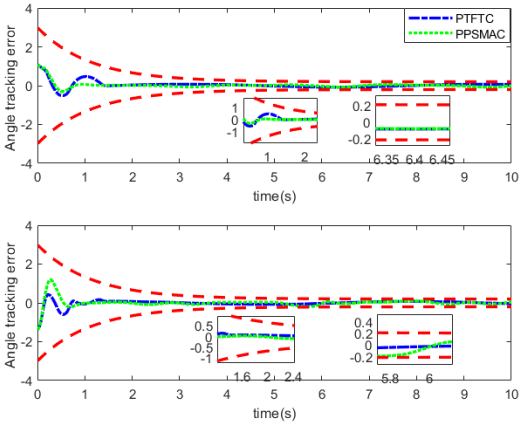


Fig. 18. Angle tracking error under different controllers.

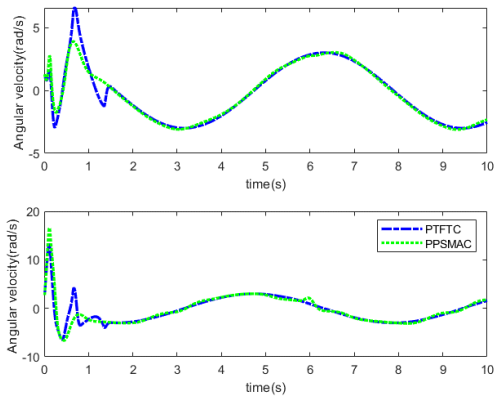


Fig. 19. Angular velocity tracking curve under different controllers.

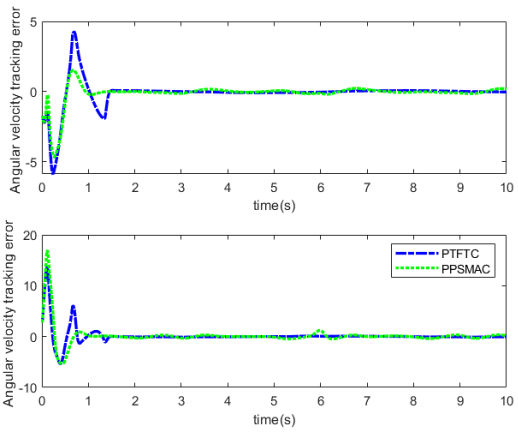


Fig. 20. Angular velocity tracking error under different controllers.

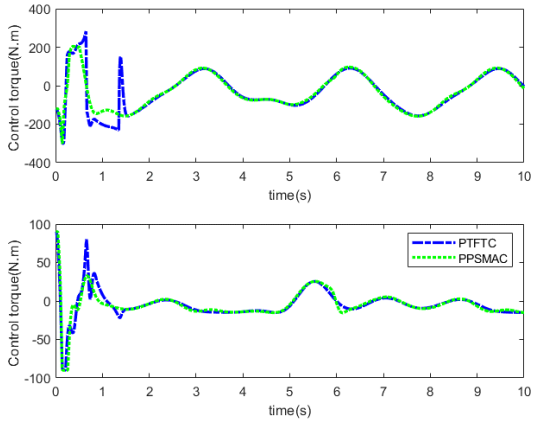


Fig. 21. Actual control input under different controllers.

Fig. 17 and Fig. 18 illustrate that both controllers effectively ensure that the angle tracking error curve of the manipulator evolves within the boundaries of the PPF, swiftly converging towards zero and accurately tracking the desired signal. Fig. 18 reveals that under the control of PPSMAC, although the angle tracking error of the manipulator is stable within the PPF, fluctuations are present. On the other hand, the PTFTC ensures that once the angle tracking error of the manipulator reaches stability, the trajectory tracking error curve takes on the form of a straight line. The rate of convergence for the angle tracking error using the PPSMAC is quicker in comparison with that of the PTFTC. Similarly, angular velocity tracking curve for both controllers is demonstrated in Fig. 19. However, the convergence time of angle tracking error with the PPSMAC is unpredictable, while the convergence time with the PTFTC can be predetermined, enabling better control over the convergence time. Fig. 20 demonstrates that under both controllers, the angular velocity tracking error of the manipulator stabilizes within 1.5 seconds. However, the angular velocity tracking error based on the PPSMAC still fluctuates to a certain extent and fails to achieve complete stability. Fig. 21 shows that the actual control torque trend of the two controllers is almost the same. The main reason for the similar control effect of the two controllers is that no matter how big the control input value is in the theoretical calculation, the actual input torque of the two control algorithms to the manipulator system is similar after the control torque limiting and failure calculation.

The comparative simulation of the two control strategies highlights the following observations: when the initial angle of the manipulator falls outside the boundaries of the prescribed performance constraint function, only the proposed controller can achieve stability in the angle tracking error. However, if the initial angle of the manipulator is situated within the defined performance function, both controllers are effective in ensuring that the angle tracking error of the manipulator reaches stability. The controller designed in this paper has a convergence time that can be predetermined, which allows for better control over the convergence time for both the angle tracking error and the angular velocity tracking error. Additionally, the controller leads to smaller steady-state errors for both angle tracking error and angular velocity tracking error.

6. CONCLUSION

Based on the tracking error conversion function and PTC theory, a fault-tolerant control strategy with PCT and prescribed performance is designed for second-order MIMO-NLS with unknown initial state value, parameter perturbation, external disturbance, actuator fault, and control saturation constraints.

(1) A performance-constrained fault-tolerant control strategy was developed, adjusting the tracking error to meet the desired performance criteria after a specified time period and guaranteeing transient performance. This approach resolves the issue of undetermined performance constraint parameters due to unknown initial state values in performance-preset

control, thus broadening the applicability of the performance-prescribed control method.

(2) The utilization of a speed function enables the establishment of a Lyapunov stability criterion for the convergence time that can be set at will. The theoretical validity of this criterion is established, thereby enhancing the theoretical foundation of PTC.

(3) The overamplitude saturation variable is introduced to convert the control input with actuator fault and control saturation constraints into unconstrained control input. An adaptive fuzzy system is designed to approximate the compound interference. A fault tolerant controller with PTC based on adaptive fuzzy system is presented to ensure that the error of nonlinear system converges to any bounded range within the predefined time. The passive fault tolerance of nonlinear system to actuator fault and the strong robustness to uncertain factors are enhanced, and the application range of fault tolerant controller is expanded.

Although the fault-tolerant controller in the current paper enhances the fault-tolerant ability and robustness of the nonlinear system, the controller designed in this paper contains the nominal variable information of the nonlinear system. Under the current development trend of artificial intelligence, establishing a model-free adaptive fault-tolerant controller and enhancing the universality and self-tuning of the controller will be the primary development trend in the future.

CONFLICT OF INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

REFERENCES

[1] K. Shao and J. Zheng, "Predefined-time sliding mode control with prescribed convergent region," *IEEE/CAA Journal of Automatica Sinica,* vol. 9, no. 5, pp. 934-936, 2022.

[2] J. D. Sánchez-Torres, A. J. Muñoz-Vázquez, M. Defoort, R. Aldana-López, and D. Gómez-Gutiérrez, "Predefined-time integral sliding mode control of second-order systems," *International Journal of Systems Science,* vol. 51, no. 16, pp. 3425-3435, 2020.

[3] N. Han, X. Ren, C. Zhang, and D. Zheng, "Prescribed performance control with input indicator for robot system based on spectral normalized neural networks," *Neurocomputing,* vol. 492, pp. 201-210, 2022.

[4] H. Lv, W. Xiang, and J. Zhu, "Finite Time Prescribed Performance Control for Uncertain Second-Order Nonlinear Systems," *Journal of Mathematics,* vol. 2022, 2022.

[5] Z. Wang, X. Liu, and W. Wang, "Adaptive prescribed performance control with selected transient response for a class of nonlinear systems with uncertainties," *International Journal of Adaptive Control and Signal Processing,* vol. 36, no. 3, pp. 670-689, 2022.

[6] N. Zhang, S. Wang, Y. Hou, and L. Zhang, "A robust predefined-time stable tracking control for uncertain robot manipulators," *IEEE Access,* vol. 8, pp. 188600-188610, 2020.

[7] A. J. Muñoz-Vázquez, J. D. Sánchez-Torres, S. Gutiérrez-Alcalá, E. Jiménez-Rodríguez, and A. G. Loukianov, "Predefined-time robust contour tracking of robotic manipulators," *Journal of the Franklin Institute,* vol. 356, no. 5, pp. 2709-2722, 2019.

[8] Y. Su, C. Zheng, and P. Mercorelli, "Robust approximate fixed-time tracking control for uncertain robot manipulators," *Mechanical Systems and Signal Processing,* vol. 135, p. 106379, 2020.

[9] J. Obregón-Flores, G. Arechavaleta, H. M. Becerra, and A. Morales-Díaz, "Predefined-time robust hierarchical inverse dynamics on torque-controlled redundant manipulators," *IEEE Transactions on Robotics,* vol. 37, no. 3, pp. 962-978, 2021.

[10] A. T. Vo, T. N. Truong, and H.-J. Kang, "A novel prescribed-performance-tracking control system with finite-time convergence stability for uncertain robotic manipulators," *Sensors,* vol. 22, no. 7, p. 2615, 2022.

[11] J. Wang, Y. Zhang, and R. Zhang, "Prescribed-time Control Based on Backstepping Method for Trajectory Tracking of Robotic Manipulators," in *2022 41st Chinese Control Conference (CCC)*, 2022, pp. 540-545: IEEE.

[12] X. Bu, B. Jiang, and H. Lei, "Low-complexity fuzzy neural control of constrained waverider vehicles via fragility-free prescribed performance approach," *IEEE Transactions on Fuzzy systems,* 2022.

[13] X. Bu, B. Jiang, and H. Lei, "Performance guaranteed finite-time non-affine control of waverider vehicles without function-approximation," *IEEE Transactions on Intelligent Transportation Systems,* vol. 24, no. 3, pp. 3252-3262, 2022.

[14] Y. Song, Y. Wang, and C. Wen, "Adaptive fault-tolerant PI tracking control with guaranteed transient and steady-state performance," *IEEE Transactions on automatic control,* vol. 62, no. 1, pp. 481-487, 2016.

[15] V. H. Pham and H.-S. Ahn, "Distributed least square solution method to linear algebraic equations over multiagent networks," *arXiv preprint arXiv:2303.07884,* 2023.

[16] H. Nguyen, B. Nguyen, H.-G. Lee, and H.-S. Ahn, "Encrypted Observer-based Control for Linear Continuous-Time Systems," *arXiv preprint arXiv:2303.00963,* 2023.

[17] C. Yu, J. Jiang, Z. Zhen, A. K. Bhatia, and S. Wang, "Adaptive backstepping control for air-breathing hypersonic vehicle subject to mismatched uncertainties," *Aerospace Science and Technology,* vol. 107, p. 106244, 2020.

[18] S. Wang, Y. Cao, T. Huang, Y. Chen, P. Li, and S. Wen, "Sliding mode control of neural networks via continuous or periodic sampling event-triggering algorithm," *Neural Networks,* vol. 121, pp. 140-147, 2020.

[19] S. Wang, Q. Chen, X. Ren, and H. Yu, "Neural network-based adaptive funnel sliding mode control for servo mechanisms with friction compensation," *Neurocomputing,* vol. 377, pp. 16-26, 2020.

[20] S. Sui, S. Tong, and Y. Li, "Observer-based fuzzy adaptive prescribed performance tracking control for nonlinear stochastic systems with input saturation," *Neurocomputing,* vol. 158, pp. 100-108, 2015.

[21] F. Chang and C. Li, "An Extended Looped Functional Approach for Stability Analysis of TS Fuzzy Impulsive Control Systems," *International Journal of Control, Automation and Systems,* pp. 1-13, 2023.

[22] C. Qian, S. Chen, C. Hua, and K.-S. Park, "Fixed-time Prescribed Performance Vibration Suppression Control of the Electromechanical Transmission System with Actuator Faults," *International Journal of Control, Automation and Systems,* vol. 21, no. 5, pp. 1431-1441, 2023.

[23] X. Zhou, C. Gao, Z.-g. Li, X.-y. Ouyang, and L.-b. Wu, "Observer-based adaptive fuzzy finite-time prescribed performance tracking control for strict-feedback systems with input dead-zone and saturation," *Nonlinear Dynamics,* vol. 103, pp. 1645-1661, 2021.

[24] P. Li, X. Zhao, H. Qin, Z. Wang, and B. Niu, "Dynamic event-triggered finite-time H∞ tracking control of switched LPV aero-engine models," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 69, no. 3, pp. 1114-1118, 2021.

[25] H. Xu, D. Yu, S. Sui, Y.-P. Zhao, C. P. Chen, and Z. Wang, "Nonsingular practical fixed-time adaptive output feedback control of MIMO nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems,* 2022.

[26] J. Tan, Y. Dong, P. Shao, and G. Qu, "Anti-saturation adaptive fault-tolerant control with fixed-time prescribed performance for UAV under AOA asymmetric constraint," *Aerospace Science and Technology,* vol. 120, p. 107264, 2022.

[27] Y. Liu, H. Li, R. Lu, Z. Zuo, and X. Li, "An overview of finite/fixed-time control and its application in engineering systems," *IEEE/CAA Journal of Automatica Sinica,* vol. 9, no. 12, pp. 2106-2120, 2022.

[28] S. Riaz, C.-W. Yin, R. Qi, B. Li, S. Ali, and K. Shehzad, "Design of Predefined Time Convergent Sliding Mode Control for a Nonlinear PMLM Position System," *Electronics,* vol. 12, no. 4, p. 813, 2023.

[29] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance," *IEEE Transactions on Automatic Control,* vol. 53, no. 9, pp. 2090-2099, 2008.

[30] L. Fu, R. Ma, H. Pang, and J. Fu, "Predefined-time tracking of nonlinear strict-feedback systems with time-varying output constraints," *Journal of the Franklin Institute,* vol. 359, no. 8, pp. 3492-3516, 2022.

[31] Y. Cao, C. Wen, and Y. Song, "Prescribed performance control of strict-feedback systems under actuation saturation and output constraint via event-triggered approach," *International Journal of Robust and Nonlinear Control,* vol. 29, no. 18, pp. 6357-6373, 2019.

[32] B. Ren, S. S. Ge, K. P. Tee, and T. H. Lee, "Adaptive neural control for output feedback nonlinear systems using a barrier Lyapunov function," *IEEE Transactions on Neural Networks,* vol. 21, no. 8, pp. 1339-1345, 2010.

[33] J. Liu, "Robot control system design and matlab simulation," *Beijing: Press of Tsinghua University,* 2008.

[34] C.-W. Yin *et al.*, "A Novel Predefined Time PD-Type ILC Paradigm for Nonlinear Systems," *Mathematics,* vol. 11, no. 1, p. 56, 2022.

ChunWu Yin received the B.S. degree in applied mathematics from Xiaogan University, Xiaogan, Hubei, China, in 2004; the M.S. degree in applied mathematics from Xi'an University of Technology, Xi'an, Shaanxi, China, in 2007; the Ph.D. in navigation guidance and control from Northwestern Polytechnical University, Xi'an, Shaanxi, China, in 2016. He is currently an Associate Professor at the School of Information and Control Engineering, Xi'an University of Architecture and Technology. His research interests include robust control, spacecraft attitude tracking control theory, advanced control theory, robotics systems control, MATLAB simulation, fractional order control, mobile robotics, and UAV multi-body dynamics fault diagnosis.

Saleem Riaz received his B.S. degree in electrical engineering from the University of Engineering and Technology (UET), Lahore, Punjab, Pakistan, in 2013. He received his M.S. and Ph.D. degrees from Northwestern Polytechnical University in 2017 and 2022, respectively. His research interests include iterative learning for nonlinear dynamical control systems, optimal control theory, and power systems.

Ali Arshad Uppal received his Ph.D. from Capital University of Science and Technology, Islamabad, Pakistan, from May 2012 to January 2016. He has worked as a Visiting Scholar at the Department of Electrical and Computer Engineering, The Ohio State University (OSU), Columbus, OH, USA. He has worked on the model-based control of the UCG process. He also worked as postdoctoral researcher at the Department of Electrical and Computer Engineering, Universidad do Porto, Porto, Portugal, from 22 September 2020 to 15 September 2021. Currently, he is an Assistant Professor at the Department of Electrical and Computer Engineering, COMSATS University Islamabad (CUI), Islamabad, Pakistan. He has completed his Ph.D. at CUI in 2016. His research interests include nonlinear control, sliding mode control, process control, numerical optimal control, and control of infinite-dimensional systems.

Jamshed Iqbal is currently working as a Senior Lecturer at the University of Hull, UK. Before joining the University of Hull, he was working as a Professor at the University of Jeddah, KSA. He holds Ph.D. in robotics from the Italian Institute of Technology (IIT) and three M.S. degrees in various fields of engineering from Finland, Sweden, and Pakistan. With more than 20 years of multi-disciplinary experience in industry and academia, his research interests include mechatronics and control systems. He has more than 120 journal papers on his credit with an H-index of 39. He is a senior member of IEEE USA and a Fellow of HEA, UK.