Reinforcement Learning-Based Prescribed Performance Neuro-Optimal Control for Robot Manipulators Under Composite Actuator Faults

G. Narayanan, Sai Navaneet, Sangtae Ahn, Sangmoon Lee*

***Abstract*—Robotic manipulators operating under unknown nonlinear dynamics, time-varying disturbances, and joint-level actuator or system faults present significant challenges in achieving reliable, robust, and high-precision motion control. To address these challenges, this article develops a reinforcement-learning-based neuro-optimal control scheme for uncertain robotic manipulators subject to composite actuator fault modes that guarantees prescribed tracking performance. The proposed control design explicitly considers actuator total loss of effectiveness (TLOE) and partial loss of effectiveness (PLOE) and compensates for abrupt system faults affecting joint dynamics through a robust control mechanism. A prescribed performance function (PPF) with a filtered variable is employed to achieve predefined-time tracking by constructing an error transformation that converts the constrained tracking problem into an unconstrained form. Unlike existing methods, this approach eliminates initial-value restrictions by ensuring that both the PPF initial condition and the transformation parameter are independent of the initial tracking error and system dynamics. Moreover, a reliable control mechanism reduces the impact of actuator faults while compensating for neural network approximation errors, with the objective functions associated with Hamilton-Jacobi-Bellman approximation errors minimized through a reinforcement-learning-based identifier-critic-actor framework. Finally, simulations based on an actual two-link manipulator model validate the superiority of the proposed control strategy, and its effectiveness is further demonstrated through comprehensive comparative studies.**

***Index Terms*—Robotic manipulators, reinforcement learning, actuator faults, prescribed performance function, predefined-time tracking.**

I. Introduction

THE advancement of robotic systems has stimulated extensive research on high-precision tracking control of uncertain robotic manipulators for industrial applications [1], [2], [3]. Since faults may cause environmental damage or security risks, robots must be equipped with fault-tolerant capabilities, especially in long-term human–robot interaction. Among all components, actuators are particularly vulnerable due to continuous heavy operation. From a control perspective, actuator faults are classified as total loss of effectiveness (TLOE) and partial loss of effectiveness (PLOE). PLOE causes performance degradation at one or more joints, and the more severe TLOE results in complete loss of actuator control. If not addressed, both can seriously impair task execution. In recent years, many fault-tolerant control (FTC) schemes have been proposed for robotic systems [4], [5] to address actuator failures caused by voltage drops or load variations that lead to PLOE or TLOE and degrade tracking performance. To ensure stability and reliability, many FTC methods have been developed to preserve functionality and reduce performance loss [6], [7]. For example, a singularity-free robust FTC was proposed in [8] to address PLOE under uncertainties and disturbances. In [9], an adaptive estimator-based FTC was designed to compensate for actuator failures, false data injection attacks, and joint friction, while [10] presented an FTC strategy for manipulators with uncertainties and unknown loads.

Recently, various control strategies have been incorporated into FTC frameworks to improve the tracking performance of robotic manipulators under uncertain dynamics and external disturbances, including robust control [11], sliding-mode control [12], neural network (NN)–based control [13], and fuzzy control [14]. To enhance convergence, an adaptive super-twisting global integral terminal sliding mode control scheme was proposed in [15], achieving finite-time convergence and improved robustness against uncertainties and actuator faults over conventional asymptotic approaches. In addition, a barrier Lyapunov function–based adaptive fuzzy finite-time FTC strategy was developed in [16] to handle actuator faults and unknown dead zones. Notably, the convergence time of most finite-time methods depends on initial conditions. To address this issue, fixed-time stability was studied in [17], where the settling time is independent of initial conditions, and a fixed-time adaptive FTC scheme based on a disturbance observer was further presented in [18].

With the growing demand for high-precision motion control in robotic manipulators, PPC has gained considerable attention for guaranteeing desired tracking behavior during both transient and steady-state phases [19], [20], [21]. By introducing prescribed performance functions (PPFs) and transforming tracking errors into constrained coordinates, PPC enforces user-defined bounds on overshoot, convergence rate, and steady-state accuracy, which is critical for high-precision applications. Although PPC has been widely studied for general nonlinear systems, its application to robotic manipulators remains limited due to high degrees of freedom, strong cou-

This work was supported in part by the Regional Innovation System & Education (RISE) Glocal 30 Program through the Daegu RISE Center, funded by the Ministry of Education (MOE) and the Daegu Metropolitan City, Republic of Korea, under Grant 2025-RISE-03-001. G. Narayanan is with the Institute for Advanced Technology Convergence, Kyungpook National University, Daegu 41566, South Korea (e-mail: narayananknu@knu.ac.kr). Sai Navaneet, Sangtae Ahn, and Sangmoon Lee are with the School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, South Korea (e-mail: sainavaneet76@gmail.com, stahn@knu.ac.kr, moony@knu.ac.kr. *Corresponding author: Sangmoon Lee.

pling, and external disturbances. Within this framework, robust adaptive PPC schemes have been developed for rigid–flexible coupled manipulators to achieve satisfactory tracking and vibration suppression under input quantization [22]. Adaptive PPC has also been applied to flexible-joint manipulators [23], while PPC-based fuzzy controllers have improved tracking performance in wheeled mobile robots [24]. Moreover, PPC-based nonfragile control strategies have been proposed to address actuator faults [25].

Recently, artificial NNs, owing to their strong universal approximation capability, have been implemented on robotic platforms, demonstrating excellent adaptive learning in handling uncertainties, input dead zones, and backlash-like hysteresis. In particular, NN-based sliding-mode control has been realized on KINOVA JACO2 robots to compensate for uncertainties and actuator dead zones [26], and an adaptive radial basis function-NN controller has been implemented on a Baxter robot to approximate unknown dynamics [27]. With increasing industrial energy demands, energy-efficient operation has become a key concern, and reinforcement learning (RL)-based neuro-optimal control offers an effective framework for achieving near-optimal performance with low resource consumption, with successful applications in unmanned surface vehicles [28], mobile robots [29], and manipulators [30]. The actor–critic–identifier framework enables simultaneous learning of system dynamics and control policies under uncertainties and actuator faults [31]. In [32], an RL-based FTC was proposed for nonlinear systems with actuator faults, and [33] developed an adaptive finite-time optimal control scheme for flexible-joint robots. However, an RL-based prescribed-performance fault-tolerant tracking framework for robotic manipulators that addresses actuator fault compensation and ensures optimal tracking remains an open problem.

Motivated by these challenges, this article proposes a PPC-based fault-tolerant tracking control framework for robotic systems using adaptive NN-based RL. The main contributions are summarized as follows:

1) Compared with existing RL-based controllers [27], [31], [32], which typically do not consider PPC or multiple actuator fault modes, this study proposes an adaptive NN-based RL framework that, for the first time, addresses PPC-based fault-tolerant tracking control of uncertain robotic systems under both PLOE and TLOE actuator faults.

2) We propose a novel performance and error transformation function that, unlike existing optimal FTC approaches [10], [11], [13], [14], [18], [19], [32], enables user-defined settling time independent of initial conditions and system parameters. Moreover, our method eliminates a common assumption in most PT control studies [3], [20], [22], [25], where the performance function or error transformation parameter depends on the initial tracking error.

3) Unlike existing methods [26], [28], this work proposes an RL-based discounted performance function that combines prescribed performance and fault compensation, using Gaussian-kernel NNs with online adaptive error estimation to ensure optimal control and guaranteed tracking performance for bounded initial conditions.

II. Problem Formulation and Preliminaries

A. Dynamic Modeling of Robotic Manipulator Systems: Consider an n-link robotic manipulator whose dynamic model is given by [26]:

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta,\dot{\theta}) + \mathcal{G}(\theta) = \tau(t) + d(t) \tag{1}$$

where $\theta(t) \in \mathbb{R}^n$ denotes the joint position, with $\dot{\theta}(t) \in \mathbb{R}^n$, and $\ddot{\theta}(t) \in \mathbb{R}^n$ representing the joint velocity and acceleration, respectively. $\mathcal{M}(\theta) \in \mathbb{R}^{n\times n}$ is the symmetric positive-definite inertia matrix, $\mathcal{C}(\theta,\dot{\theta}) \in \mathbb{R}^{n\times n}$ denotes the Coriolis and centrifugal terms, and $\mathcal{G}(\theta) \in \mathbb{R}^n$ is the gravity vector. The control input is $\tau(t) \in \mathbb{R}^n$, and $d(t) \in \mathbb{R}^n$ is a bounded disturbance satisfying $\|d(t)\| \le \bar{d}$, where $\bar{d} > 0$.

To account for model uncertainties, the system matrices are decomposed into nominal parts $\mathcal{M}_0(\theta), \mathcal{C}_0(\theta,\dot{\theta})$, and $\mathcal{G}_0(\theta)$, and uncertain parts $M_\Delta(\theta)$, $\mathcal{C}_\Delta(\theta,\dot{\theta})$, and $\mathcal{G}(\theta)$, such that $\mathcal{M}(\theta) = \mathcal{M}_0(\theta) + \mathcal{M}_\Delta(\theta)$, $\mathcal{C}(\theta,\dot{\theta}) = \mathcal{C}_0(\theta,\dot{\theta}) + \mathcal{C}_\Delta(\theta,\dot{\theta})$, and $\mathcal{G}(\theta) = \mathcal{G}_0(\theta) + \mathcal{G}_\Delta(\theta)$.

Since the nominal inertia matrix $\mathcal{M}_0(\theta)$, is nonsingular, the dynamic model in (1) can be reformulated as

$$\ddot{\theta} = M_0^{-1}(\theta)\tau(t) + M_0^{-1}(\theta)d(t) + \ell(t), \tag{2}$$

where the lumped uncertainty term $\ell(t)$ is $\ell(t) = \mathcal{M}^{-1}(\theta)(-\mathcal{C}(\theta,\dot{\theta})\dot{\theta} - \mathcal{G}(\theta)) + (\mathcal{M}^{-1}(\theta) - M_0^{-1}(\theta))(\tau(t) + d(t))$. Let $\delta_1 = \theta = [\delta_{11},\ldots,\delta_{1n}]^T$ and $\delta_2 = \dot{\theta} = [\delta_{21},\ldots\delta_{2n}]^T$. Accordingly, (2) is reformulated as follows:

$$\begin{aligned}\dot{\delta}_1(t) =& \delta_2(t),\\ \dot{\delta}_2(t) =& M_0^{-1}(\theta)\tau(t) + M_0^{-1}(\theta)d(t) - \ddot{q}_d(t) + \ell(t).\end{aligned} \tag{3}$$

Considering faults that may occur in the driving motors or their associated gear trains due to actuator aging [11], the actual control torque of the manipulator can be modeled as

$$\tau_\alpha(t) = \bar{\xi}_{\alpha\beta}(t)\,\zeta_{\alpha\beta}(t)\,\vartheta_\alpha(t) + \gamma_{\alpha\beta}(t), \tag{4}$$

where $\alpha = 1,\ldots,n$ and $\beta = 1,2,\ldots$, with n denoting the number of actuators; $\tau_\alpha(t)$ and $\vartheta_\alpha(t)$ represent the actual output torque and the commanded input torque of the α-th, respectively; $\xi_{\alpha\beta}(t) \in [0,1]$ captures partial degradation, $\gamma_{\alpha\beta}(t)$ represents an unknown bounded additive fault, and $\zeta_{\alpha\beta}(t) \in \{0,1\}$ is a binary switching variable. Specifically, $\zeta_{\alpha\beta}(t) = 0$ indicates that the input torque $\vartheta_\alpha(t)$ is completely blocked and does not influence the manipulator dynamics, whereas $\zeta_{\alpha\beta}(t) = 1$ indicates that the actuator is active and transmits torque to the joint. In summary, all actuator operating modes, including normal operation, PLOE (LOE, bias), and TLOE (stuck or outage), are listed in Table I.

TABLE I
Classification of Actuator Operating and Fault Modes

Actuator Mode	$\xi_{\alpha\beta}(t)$	$\gamma_{\alpha\beta}(t)$
Normal	$\xi_{\alpha\beta}(t) = 1$	$\gamma_{\alpha\beta} = 0$
Bias	$\xi_{pq}(t) = 1$	$\gamma_{\alpha\beta}(t) \neq 0$
Stuck	$\xi_{\alpha\beta}(t) = 0$	$\gamma_{\alpha\beta}(t) = \gamma_{\alpha\beta}(t_\beta)$
Outage	$\xi_{\alpha\beta}(t) = 0$	$\gamma_{\alpha\beta}(t) = 0$
LOE	$0 < \xi_{\min} \le \xi_{\alpha\beta}(t) < 1$	$\gamma_{\alpha\beta}(t) = 0$

Define the vectors $\vartheta(t) = [\vartheta_1(t),\ldots,\vartheta_n(t)]^T$, $\xi_\beta(t) =$

$\text{diag}\{\bar{\xi}_{1\beta}(t),\ldots,\bar{\xi}_{n\beta}(t)\}$, $\bar{\zeta}_\beta(t) = \text{diag}\{\zeta_{1\beta}(t),\ldots,\zeta_{n\beta}(t)\}$, $\gamma_\beta(t) = [\gamma_{1\beta}(t),\ldots,\gamma_{n\beta}(t)]^T$. Then, the fault model is

$$\tau(t) = \xi_\beta(t)\bar{\zeta}_\beta(t)\vartheta(t) + \gamma_\beta(t). \tag{5}$$

To facilitate controller design and stability analysis, the following assumptions are imposed.

Assumption 1: The reference signal $\theta_d(t)$ and its derivative $\dot{\delta}_d(t)$ are known, continuous, and bounded.

Assumption 2: The external disturbance $d(t)$ and its derivative $\dot{d}(t)$ are unknown but bounded.

Remark 1: In robotic manipulators, the actual output torque $\tau(t)$ captures partial degradation, bounded faults, and actuator activation, respectively. Existing studies [13], [14], [18], [32] are limited to narrow fault ranges and rarely consider repeated or stochastic faults. The generalized actuator failure model (5) captures multiple, arbitrary fault occurrences, β, providing a realistic representation of actuator behavior. By explicitly modeling the control torque actuator and leveraging redundancy, multi-actuator systems, e.g., multicylinder hydraulic manipulators and single-link servo-driven arms, maintain robust, continuous, and reliable operation under both conventional and extended faults.

B. Predefined Time Error Transformation Function: A smooth PPF is designed, extending the approaches in [20], [23], to ensure predefined convergence of the tracking error. Unlike traditional bounds that depend on the initial error, the PPF imposes a time-varying constraint that monotonically decreases to a steady limit within a user-defined settling time. The PPF is defined as

$$\aleph(t) = \begin{cases} \varkappa_a \left(\dfrac{\mathcal{T}_d - t}{\mathcal{T}_d}\right)^{\varkappa_c} + \varkappa_b, & t \in [0, \mathcal{T}_d), \\ \varkappa_b, & t \in [\mathcal{T}_d, +\infty), \end{cases} \tag{6}$$

where $\mathcal{T}_d > 0$ specifies the prescribed settling time, and $\varkappa_a, \varkappa_b, \varkappa_c > 0$ represent design constants satisfying $\varkappa_a + \varkappa_b = \frac{\pi}{2}$, which ensures $\aleph(0) = \frac{\pi}{2}$, which defines a tunable convergence profile smoothly decreasing from a large initial bound to a tighter steady-state range.

Remark 2: The PPF $\aleph(t)$ in (6) imposes a time-varying bound on the tracking error that is independent of the initial condition. During the interval $t \in [0, \mathcal{T}_d)$, $\aleph(t)$ decreases according to the shaping parameter $\varkappa_c$, forcing the error to evolve within a shrinking boundary and ensuring convergence within the prescribed time $\mathcal{T}_d$. This mechanism alleviates the initial error constraint while regulating the transient behavior, and for $t \geq \mathcal{T}_d$, $\aleph(t)$ becomes constant at $\varkappa_b$, thereby fixing the steady-state error bound and transitioning the system into steady-state tracking.

For all $t \geq 0$, the constructed function $\aleph(t)$ exhibits the following properties: (1) $\varkappa_b \leq \aleph(t) \leq \frac{\pi}{2}$ and $\dot{\aleph}(t) \leq 0$ ensuring a monotonically decreasing performance boundary; (2) $\aleph(t)$ is continuously differentiable, and both $\aleph(t)$ and $\dot{\aleph}(t)$ bounded. The constrained tracking error ε_1 is mapped to an unconstrained form for control design:

$$\varepsilon_1 = \tan\left(\frac{\pi}{2\aleph(t)}\arctan(\wp_1)\right). \tag{7}$$

Using (6) and (7), the initial condition satisfies $\varepsilon_1(0) = \wp_1(0)$ for any $\wp_1(0) \in \mathbb{R}$, ensuring that the transformation does not introduce discontinuities. Moreover, if it ε_1 remains bounded for $t \geq 0$, the following relationship holds: $|\wp_1| < \tan(\aleph(t)),\ t \geq 0, \quad |\wp_1| < \tan(\varkappa_b),\ t \geq \mathcal{T}_d$. Thus, ensuring the boundedness of ε_1 guarantees that the tracking error meets the prescribed performance, converging within the predefined time without overshoot, while reducing the constrained control problem to regulating an unconstrained transformed error.

By inverting the transformation in (7), the transformed error can be reconstructed as

$$\wp_1 = \tan\left(\frac{2\aleph(t)}{\pi}\arctan(\varepsilon_1)\right). \tag{8}$$

Differentiating (8) with respect to time yields the dynamics of the transformed error:

$$\dot{\wp}_1 = \frac{2\aleph(t)}{\pi}\frac{\dot{\varepsilon}_1}{(1+\varepsilon_1^2)\cos^2\left(\frac{2\aleph(t)}{\pi}\arctan(\varepsilon_1)\right)} + \frac{2\dot{\aleph}(t)}{\pi} \times \frac{\arctan(\varepsilon_1)}{\cos^2\left(\frac{2\aleph(t)}{\pi}\arctan(\varepsilon_1)\right)}. \tag{9}$$

Consequently, the transformed error dynamics can be expressed in compact form as

$$\dot{\wp}_1 = \varkappa_\sigma \dot{\varepsilon}_1 - \hbar_{d1}, \tag{10}$$

where $\varkappa_\sigma = \frac{2\aleph(t)}{\pi}(1+\varepsilon_1^2)^{-1}\cos^{-2}\left(\frac{2\aleph(t)}{\pi}\arctan(\varepsilon_1)\right)$, and $\hbar_{d1} = \frac{2\dot{\aleph}(t)}{\pi}\arctan(\varepsilon_1)\cos^{-2}\left(\frac{2\aleph(t)}{\pi}\arctan(\varepsilon_1)\right)$. Equation (10) demonstrates that the derivative of the unconstrained transformed error $\wp_1$ to the original tracking error derivative $\dot{\varepsilon}_1$, while $\hbar_{d1}$ accounts for the effect of the time-varying performance boundary $\aleph(t)$.

Remark 3: In robotic manipulator systems, achieving prescribed performance is often complicated by actuator faults, joint limits, payload variations, and external disturbances. Existing prescribed-time (PT) control methods [3], [21], [22], [25] typically require initial value constraints on the performance function or error transformation, e.g., $|\wp_1(0)| < \aleph(0)$ or $|\wp_1(0)| < \bar{\Psi}\aleph(0)$, which limits their applicability when initial states are unknown or inaccessible. In contrast, the proposed predefined-time error transformation maps the constrained tracking error ϵ_1 to an unconstrained variable $\wp_1$ via a smooth, tunable performance function $\aleph(t)$ with parameters $\varkappa_a$, $\varkappa_b$, $\varkappa_c$, and prescribed settling time $\mathcal{T}_d$. Compared with previous studies [3], [21], [22], [25], this design ensures that the tracking error remains within bounds, converges monotonically within $\mathcal{T}_d$, maintains bounded dynamics, and guarantees overshoot-free, robust, and practically implementable performance.

C. Radial Basis Function Neural Networks: Radial basis function NNs are widely employed to approximate unknown nonlinear continuous functions due to their universal approximation capability [12], [19], [26]. Consider a continuous unknown function $g(\bar{\varkappa})$ defined over a compact set $\mathbb{U}$. Then, for any $\kappa > 0$, there exists NN $\Theta^T\mathcal{Q}(\bar{\varkappa})$ such that the approximation error is bounded as $\sup_{\bar{\varkappa}\in\mathbb{U}}\left|g(\bar{\varkappa}) - \Theta^T\mathcal{Q}(\bar{\varkappa})\right| < \kappa$, where $\bar{\varkappa} = [\bar{\varkappa}_1, \bar{\varkappa}_2, \ldots, \bar{\varkappa}_n]^T \in \mathbb{U} \subset \mathbb{R}^n$ is the input vector, $\Theta = [\Theta_1, \Theta_2, \ldots, \Theta_r]^T \in \mathbb{R}^r$ is the optimal weight vector,

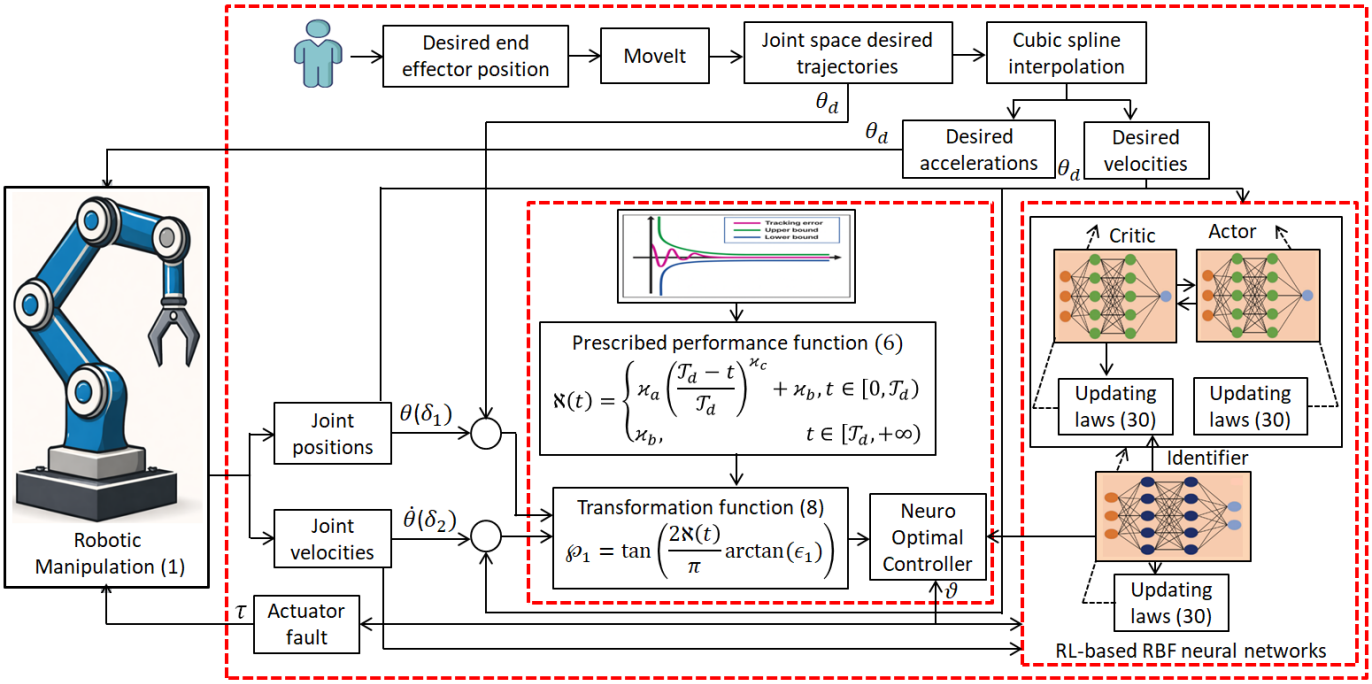

Fig. 1. Schematic of the proposed PPC-based fault-tolerant tracking control framework for robotic systems using RL-based radial basis function (RBF) NN.

and r is the number of neurons. The basis function vector $\mathcal{Q}(\bar{\varkappa})$ is chosen as a Gaussian function:

$$\mathcal{Q}(\bar{\varkappa}) = \exp\left\{-\frac{(\bar{\varkappa} - q_\varsigma)^T(\bar{\varkappa} - q_\varsigma)}{\bar{b}_\varsigma^2}\right\}, \ \varsigma = 1, \ldots, r, \quad (11)$$

where $q_\varsigma = [q_{\varsigma 1}, q_{\varsigma 2}, \ldots, q_{\varsigma n}]$ denotes the center of the receptive field and $\bar{b}_\varsigma$ is the width of the Gaussian function.

D. Control Objective: The objective of this study is to design a PPC-based fault-tolerant tracking control for the robotic system (3) using adaptive NN-based RL such that: (1) The tracking error $\wp_1$ converges to a predefined region within a prescribed time, satisfying $|\wp_1| < \tan(\aleph)$ for $0 \leq t \leq \mathcal{T}_d$ during the transient phase and $|\wp_1| < \tan(\varkappa_b)$ for $t \geq \mathcal{T}_d$ at steady state; (2) The cost function in (12) is minimized; and (3) All the closed-loop signals are bounded, and the tracking error remains within the prescribed performance bounds

Moreover, Fig. 1 illustrates the proposed control method, providing a clear overview of its structure.

III. MAIN RESULTS

In this section, a neuro-adaptive predefined-time backstepping controller is proposed to improve actuator fault tolerance, and an identifier-critic-actor RL framework is developed to achieve FTC with the desired performance.

Step 1: For system (3), define the tracking error $\varepsilon_1 = \delta_1 - \theta_d$ and define the optimal performance index for the performance-related error $\wp_1$ as

$$\Xi_1(\wp_1, \aleph) = \int_t^\infty \bar{\chi}_1(\wp_1(s), \aleph(s), \varphi_1(s))\, ds, \quad (12)$$

where $\bar{\chi}_1(\cdot) = \tan(\frac{2\aleph}{\pi}\arctan(\wp_1))^T \tan(\frac{2\aleph}{\pi}\arctan(\wp_1)) + \varphi_1^T\varphi_1$. The objective is to compute the optimal control φ_1^* that minimizes the cost function, yielding

$$\Xi_1^*(\wp_1, \aleph) = \min_{\varphi_1 \in \mathbb{U}} \left\{\int_t^\infty \bar{\chi}_1(\wp_1(s), \aleph(s), \varphi_1(s))\, ds\right\}. \quad (13)$$

Based on this cost function, the Hamiltonian of the system is expressed as

$$\widehat{\mathcal{H}}_1(\wp_1, \aleph, \varphi_1, \tfrac{\partial \Xi_1^*}{\partial \wp_1}, \tfrac{\partial \Xi_1^*}{\partial \aleph}) = \bar{\chi}_1(\wp_1, \aleph, \varphi_1) + \frac{\partial \Xi_1^*}{\partial \wp_1}^T \dot{\wp}_1 + \frac{\partial \Xi_1^*}{\partial \aleph}\dot{\aleph}.$$

By enforcing the stationarity principle, the condition for optimal control is obtained [31]: $\frac{\partial \widehat{\mathcal{H}}_1}{\partial \varphi_1} = 2\varphi_1^T + \varkappa_\sigma \eta_\varsigma \frac{\partial \Xi_1^*}{\partial \wp_1^T} = 0$, which provides the optimal control input

$$\varphi_1^* = -\frac{\varkappa_\sigma \eta_\varsigma}{2}\frac{\partial \Xi_1^*}{\partial \wp_1}. \quad (14)$$

Since $\frac{\partial^2 \widehat{\mathcal{H}}_1}{\partial \varphi_1^2} = 2 > 0$ the control input in (14) minimizes $\widehat{\mathcal{H}}_1$. However, because the term $\frac{\partial \Xi_1^*}{\partial \wp_1}$ is unknown, the optimal law cannot be directly applied and is estimated via the HJB equation from Bellman's principle:

$$\begin{aligned}\mathcal{H}_1\left(\wp_1, \aleph, \varphi_1^*, \frac{\partial \Xi_1^*}{\partial \wp_1}, \frac{\partial \Xi_1^*}{\partial \aleph}\right) &= \widehat{\mathcal{H}}_1\left(\wp_1, \aleph, \varphi_1^*, \frac{\partial \Xi_1^*}{\partial \wp_1}, \frac{\partial \Xi_1^*}{\partial \aleph}\right) + \frac{\partial \Xi_1^*}{\partial t} \\ &= \tan\left(\frac{2\aleph}{\pi}\arctan(\wp_1)\right)^T \tan\left(\frac{2\aleph}{\pi}\arctan(\wp_1)\right) + \varphi_1^{*T}\varphi_1^* \\ &+ \frac{\partial \Xi_1^*}{\partial \wp_1^T}\left(\varkappa_\sigma\left(\eta_\varsigma \varphi_1^* - \dot{\hbar}_d\right) - \hbar_{d1}\right) + \frac{\partial \Xi_1^*}{\partial \aleph}\frac{d\aleph}{dt} + \frac{\partial \Xi_1^*}{\partial t} = 0. \end{aligned} \quad (15)$$

To ensure PPC, the optimal value derivative $\frac{\partial \Xi_1^*}{\partial \wp_1}$ and the corresponding control input φ_1^* are designed as

$$\frac{\partial \Xi_1^*}{\partial \wp_1} = \frac{\varkappa_\sigma^{-1}}{\eta_\varsigma^2}\left(2\varrho_{a1}\wp_1 + \Im_1(\bar{\varkappa}_1) + 2\mho_1(\bar{\varkappa}_1)\right),$$

$$\varphi_1^* = \frac{1}{\varkappa_\sigma \eta_\varsigma}\big(- \varrho_{a1}\wp_1 - \frac{1}{2}\Im_1(\bar{\varkappa}_1) - \mho_1(\bar{\varkappa}_1)\big), \tag{16}$$

where $\varrho_{a1} > 0$ is a design constant, and the auxiliary terms are defined as $\Im_1(\bar{\varkappa}_1) = -2\varrho_{a1}\wp_1 - 2\mho_1(\bar{\varkappa}_1) + \varkappa_\sigma \eta_\sigma^2 \frac{\partial \Xi_1^*}{\partial \wp_1}$, and $\mho_1(\bar{\varkappa}_1) = \varkappa_\sigma \eta_\varsigma\big(\frac{\varkappa_\sigma \eta_\varsigma \wp}{2} - \dot{\theta}_d - \hbar_{d1}\big)$ with $\varkappa_1 = [\wp_1, \aleph, \theta_d, \dot{\theta}_d]^T$. The nonlinear functions $\Im_1(\bar{\varkappa}_1)$ and $\mho_1(\bar{\varkappa}_1)$ are unknown and cannot be directly implemented, and are therefore approximated using NNs as in [12], [19]:

$$\Im_1(\bar{\varkappa}_1) = \Theta_{\Im_1}^{*T}\mathcal{Q}_{\Im_1}(\bar{\varkappa}_1) + \kappa_{\Im_1}(\bar{\varkappa}_1), \tag{17}$$

$$\mho_1(\bar{\varkappa}_1) = \Theta_{\mho_1}^{*T}\mathcal{Q}_{\mho_1}(\bar{\varkappa}_1) + \kappa_{\mho_1}(\bar{\varkappa}_1), \tag{18}$$

where $\Theta_{\Im_1}^*$ and $\Theta_{\mho_1}^*$ are the ideal weight vectors, $\mathcal{Q}_{\Im_1}(\bar{\varkappa}_1)$ and $\mathcal{Q}_{\mho_1}(\bar{\varkappa}_1)$ denote the activation functions, and $\kappa_{\Im_1}(\bar{\varkappa}_1)$ and $\kappa_{\mho_1}(\bar{\varkappa}_1)$ represent the approximation errors, satisfying $|\kappa_{\Im_1}(\bar{\varkappa}_1)| \le \bar{\kappa}_{\Im_1}$ and $|\kappa_{\mho_1}(\bar{\varkappa}_1)| \le \bar{\kappa}_{\mho_1}$.

As the ideal weights are unknown a priori, adaptive NNs are used to estimate them online. In particular, the critic network weights $\widehat{\Theta}_{\mho_1}$ and the identifier network weights $\widehat{\Theta}_{\mathcal{C}_1}$ approximate $\Theta_{\mho_1}^*$ and $\Theta_{\Im_1}^*$, yielding the implementable forms:

$$\Im_1(\bar{\varkappa}_1) = \widehat{\Theta}_{\mathcal{C}_1}^T\mathcal{Q}_{\Im_1}(\bar{\varkappa}_1), \quad \mho_1(\bar{\varkappa}_1) = \widehat{\Theta}_{\mho_1}^T\mathcal{Q}_{\mho_1}(\bar{\varkappa}_1). \tag{19}$$

The adaptive update of the identifier NN weight $\widehat{\Theta}_{\mho_1}$ is governed by the following learning law:

$$\dot{\widehat{\Theta}}_{\mho_1} = -\lambda_{\Delta 1}\widehat{\Theta}_{\mho_1} + \mathcal{Q}_{\mho_1}(\bar{\varkappa}_1)\wp_1\varkappa_{\sigma 1}^{-1}, \tag{20}$$

where $\lambda_{\Delta 1} > 0$ is a design learning gain. Subsequently, the critic NN for evaluating the control performance is constructed in accordance with (19). Using the adaptive NN weights, the unknown term $\frac{\partial \Xi_1^*}{\partial \wp_1}$ in (16) can be estimated online as:

$$\frac{\partial \widehat{\Xi}_1^*}{\partial \wp_1} = \frac{\varkappa_\sigma^{-1}}{\eta_\varsigma^2}\big(2\varrho_{a1}\wp_1 + \widehat{\Theta}_{\mathcal{C}_1}^T\mathcal{Q}_{\Im_1}(\bar{\varkappa}_1) + 2\widehat{\Theta}_{\mho 1}^T\mathcal{Q}_{\mho_1}(\bar{\varkappa}_1)\big), \tag{21}$$

where $\frac{\partial \widehat{\Xi}_1^*}{\partial \wp_1}$ denotes the estimated derivative, and $\widehat{\Theta}_{\mathcal{C}_1}$ and $\widehat{\Theta}_{\mho_1}$ are the critic and identifier NN weights, respectively.

The critic NN weight $\widehat{\Theta}_{\mathcal{C}1}$ is updated according to the following learning law:

$$\dot{\widehat{\Theta}}_{\mathcal{C}1} = -\lambda_{\mathcal{C}1}\big(\mathcal{Q}_{\Im_1}(\bar{\varkappa}_1)\mathcal{Q}_{\Im_1}^T(\bar{\varkappa}_1) + \mu_\Delta \mathcal{I}_m\big)\widehat{\Theta}_{\mathcal{C}1}\big), \tag{22}$$

where $\lambda_{\mathcal{C}1} > 0, \mu_\Delta > 0$ is a design gain. Based on (16), the optimal control policy is generated by the actor NN as

$$\widehat{\varphi}_1^* = \frac{1}{\varkappa_\sigma \eta_\varsigma}\big(- \frac{1}{2}\widehat{\Theta}_{\mathcal{A}_1}^T\mathcal{Q}_{\Im_1}(\bar{\varkappa}_1) - \widehat{\Theta}_{\mho_1}^T\mathcal{Q}_{\mho_1}(\bar{\varkappa}_1)\big), \tag{23}$$

where $\widehat{\Theta}_{\mathcal{A}_1}$ represents the actor NN weight vector updated online to achieve optimal control performance. The adaptation law for the actor NN weight vector $\widehat{\Theta}_{\mathcal{A}1}$ is given by

$$\dot{\widehat{\Theta}}_{\mathcal{A}1} = -\big(\lambda_{\mathcal{A}1}\widehat{\Theta}_{\mathcal{C}1} + \lambda_{\mathcal{A}1}(\widehat{\Theta}_{\mathcal{A}1} - \widehat{\Theta}_{\mathcal{C}1})\big) \times \big(\mathcal{Q}_{\Im_1}(\bar{\varkappa}_1)\mathcal{Q}_{\Im_1}^T(\bar{\varkappa}_1) + \mu_\Delta \mathcal{I}_m\big), \tag{24}$$

where $\lambda_{\mathcal{A}1} > 0$ is a design actor learning gain.

Step 2: Consider the performance-related error variable $\wp_2 = \delta_2 - \varphi_1$, whose dynamics are given by

$$\dot{\wp}_2 = \mathcal{M}_0^{-1}\big(\xi_\beta \bar{\zeta}_\beta \vartheta + \gamma_\beta\big) + \mathcal{M}_0^{-1}d - \ddot{q}_d + \ell - \dot{\varphi}_1. \tag{25}$$

The corresponding optimal performance function is defined as $\Xi_2(\wp_2, \aleph) = \int_t^\infty \bar{\chi}_2(\wp_2(s), \aleph(s), \vartheta(s))ds$, where $\bar{\chi}_2(\wp_2, \aleph, \vartheta) = \tan(\frac{2\aleph}{\pi}\arctan(\wp_2))^T\tan(\frac{2\aleph}{\pi}\arctan(\wp_2)) + \vartheta^{*T}\vartheta^*$. Taking the time derivative of $\Xi_2(\wp_2, \aleph)$, along the system trajectories in (25), the associated HJB equation is obtained as

$$\mathcal{H}_2\Big(\wp_2, \aleph, \vartheta^*, \frac{\partial \Xi_2^*}{\partial \wp_2}, \frac{\partial \Xi_2^*}{\partial \aleph}\Big) = \tan\big(\frac{2\aleph}{\pi}\arctan(\wp_2)\big)^T \times \tan\big(\frac{2\aleph}{\pi}\arctan(\wp_2)\big) + \vartheta^{*T}\vartheta^* + \frac{\partial \Xi_2^*}{\partial \wp_2^T}\big[\mathcal{M}_0^{-1}\big(\xi_\beta\bar{\zeta}_\beta\vartheta + \gamma_\beta\big) + \mathcal{M}_0^{-1}d - \ddot{q}_d + \ell - \dot{\varphi}_1\big] + \frac{\partial \Xi_2^*}{\partial \aleph}\frac{d\aleph}{dt} + \frac{\partial \Xi_2^*}{\partial t} = 0. \tag{26}$$

As in (14), the stationarity condition $\frac{\partial \mathcal{H}_2}{\partial \vartheta} = 0$ yields

$$\vartheta^* = -\frac{1}{2\mathcal{M}_0}\xi_\beta\bar{\zeta}_\beta\frac{\partial \Xi_2^*}{\partial \wp_2}. \tag{27}$$

To realize predefined-time FTC design, the derivative $\frac{\partial \Xi_2^*}{\partial \wp_2}$ and the corresponding optimal control input ϑ^* are constructed as

$$\frac{\partial \Xi_2^*}{\partial \wp_2} = \frac{\mathcal{M}_0}{\xi_\beta^2\bar{\zeta}_\beta^2}\Big(2\varrho_{a2}\mathcal{M}_0\wp_2 + \Im_2(\bar{\varkappa}_2) + 2\mho_2(\bar{\varkappa}_2)\Big),$$

$$\vartheta^* = \frac{1}{\xi_\beta\bar{\zeta}_\beta}\Big(\varrho_{a2}\wp_2 + \frac{\Im_2(\bar{\varkappa}_2)}{2} + \mho_2(\bar{\varkappa}_2)\Big), \tag{28}$$

where $\varrho_{a2} > 0$ is a design constant. The auxiliary nonlinear terms are defined as $\Im_2(\bar{\varkappa}_2) = -2\varrho_{a2}\mathcal{M}_0\wp_2 - 2\mathcal{M}_0^{-1}\phi_\iota\,\mathrm{sign}(\wp_2) - 2\mho_2(\bar{\varkappa}_2) + \xi_\beta\mathcal{M}_0^{-1}\bar{\zeta}_\beta^2\frac{\partial \Xi_2^*}{\partial \wp_2}$, $\mho_2(\bar{\varkappa}_2) = \ell - \ddot{q}_d - \dot{\varphi}_1$, and $\phi_\iota = |\mathcal{M}_0^{-1}\gamma_\beta|$ with $\bar{\varkappa}_2 = [\wp_2, \aleph, \delta, \widehat{\Theta}_{\mho_1}, \widehat{\Theta}_{\mathcal{A}_1}]^T$. Since the nonlinear functions $\Im_2(\bar{\varkappa}_2)$ and $\mho_2(\bar{\varkappa}_2)$ are unknown, adaptive NNs are employed to approximate them as $\Im_2(\bar{\varkappa}_2) = \Theta_{\Im_2}^{*T}\mathcal{Q}_{\Im_2}(\bar{\varkappa}_2) + \kappa_{\Im_2}(\bar{\varkappa}_2)$, $\mho_2(\bar{\varkappa}_2) = \Theta_{\mho_2}^{*T}\mathcal{Q}_{\mho_2}(\bar{\varkappa}_2) + \kappa_{\mho_2}(\bar{\varkappa}_2)$, which leads to the following implementable FTC laws:

$$\frac{\partial \widehat{\Xi}_2^*}{\partial \wp_2} = \frac{\mathcal{M}_0}{\xi_\beta^2\bar{\zeta}_\beta^2}\Big(2\varrho_{a2}\mathcal{M}_0\wp_2 + \widehat{\Theta}_{\mathcal{C}_2}^T\mathcal{Q}_{\Im_2} + 2\widehat{\Theta}_{\mho_2}^T\mathcal{Q}_{\mho_2}(\bar{\varkappa}_2)\Big),$$

$$\hat{\vartheta}^* = \frac{1}{\xi_\beta\bar{\zeta}_\beta}\Big(\varrho_{a2}\mathcal{M}_0\wp_2 + \frac{\widehat{\Theta}_{\mathcal{A}_2}^T\mathcal{Q}_{\Im_2}}{2} + \widehat{\Theta}_{\mho_2}^T\mathcal{Q}_{\mho_2}(\bar{\varkappa}_2)\Big). \tag{29}$$

The update laws for the identifier-critic-actor weight vectors $\widehat{\Theta}_{\mho\psi}$, $\widehat{\Theta}_{\mathcal{C}\psi}$ and $\widehat{\Theta}_{\mathcal{A}\psi}$ $(\psi = 1, 2)$ are designed as

$$\begin{aligned}
\dot{\widehat{\Theta}}_{\mho\psi} =& -\Psi_\psi\big(\lambda_{\Delta\psi}\widehat{\Theta}_{\mho_1} - \mathcal{Q}_{\mho_\psi}(\bar{\varkappa}_\psi)\wp_\psi\bar{\mu}_\psi\big),\\
\dot{\widehat{\Theta}}_{\mathcal{C}\psi} =& -\lambda_{\mathcal{C}\psi}\big(\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}_{\Im_\psi}^T(\bar{\varkappa}_\psi) + \mu_\Delta\mathcal{I}_m\big)\widehat{\Theta}_{\mathcal{C}\psi}\big),\\
\dot{\widehat{\Theta}}_{\mathcal{A}\psi} =& -\big(\lambda_{\mathcal{A}\psi}\widehat{\Theta}_{\mathcal{C}\psi} + \lambda_{\mathcal{A}\psi}(\widehat{\Theta}_{\mathcal{A}\psi} - \widehat{\Theta}_{\mathcal{C}\varepsilon})\big)\\
&\times\big(\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}_{\Im_\psi}^T(\bar{\varkappa}_\psi) + \mu_\Delta\mathcal{I}_m\big),
\end{aligned} \tag{30}$$

where $\bar{\mu}_\psi = \begin{cases}\varkappa_\sigma^{-1}, & \psi = 1,\\ \mathcal{M}_0, & \psi = 2.\end{cases}$, μ_Δ, $\lambda_{\Delta\psi}$, $\lambda_{\mathcal{C}\psi}$, and $\lambda_{\mathcal{A}\psi}$ are positive design parameters associated with the adaptive learning rates of the identifier–critic–actor NNs. Specifically, $\mu_\Delta > 0$, $\lambda_{\Delta\psi} > 0$, $\lambda_{\mathcal{A}\psi} > \frac{1}{2}$, and $\lambda_{\mathcal{C}\psi} > \frac{\lambda_{\mathcal{A}\psi}}{2}$. Moreover, it $\mathcal{I}_m \in \mathbb{R}^{\psi\times\psi}$ represents the identity matrix.

According to (15), the Bellman residual error can be obtained as $\mathcal{F}_\psi = \mathcal{H}_\psi\big(\wp_\psi, \aleph, \vartheta^*, \frac{\partial \widehat{\Xi}_\psi^*}{\partial \wp_\psi}, \frac{\partial \widehat{\Xi}_\psi^*}{\partial \aleph}\big) - \mathcal{H}_\psi\big(\wp_\psi, \aleph, \vartheta^*, \frac{\partial \Xi_\psi^*}{\partial \wp_\psi}, \frac{\partial \Xi_\psi^*}{\partial \aleph}\big) = \mathcal{H}_\psi\big(\wp_\psi, \aleph, \vartheta^*, \frac{\partial \widehat{\Xi}_\psi^*}{\partial \wp_\psi}, \frac{\partial \widehat{\Xi}_\psi^*}{\partial \aleph}\big)$. According to the above analysis, the optimal

virtual controller φ_ψ must satisfy $\mathcal{F}_\psi \to 0$. If $\mathcal{H}_\psi\big(\wp_\psi, \aleph, \vartheta^*, \frac{\partial \widehat{\Xi}^*_\psi}{\partial \wp_\psi}, \frac{\partial \widehat{\Xi}^*_\psi}{\partial \aleph}\big) = 0$ holds and has a unique solution, then $\partial \mathcal{H}_\psi\big(\wp_\psi, \aleph, \vartheta^*, \frac{\partial \widehat{\Xi}^*_\psi}{\partial \wp_\psi}, \frac{\partial \widehat{\Xi}^*_\psi}{\partial \aleph}\big)/\partial \widehat{\Theta}_{\mathcal{A}\psi} = \frac{1}{2}\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)(\widehat{\Theta}_{\mathcal{A}\psi} - \widehat{\Theta}_{\mathcal{C}\psi}) = 0$.

To ensure that the update laws satisfy (30), define a nonnegative function $\Lambda_\psi(t) = \mathrm{Tr}\Big((\widehat{\Theta}_{\mathcal{A}\psi} - \widehat{\Theta}_{\mathcal{C}\psi})^T(\widehat{\Theta}_{\mathcal{A}\psi} - \widehat{\Theta}_{\mathcal{C}\psi})\Big)$. Since $\frac{\partial \Lambda_\psi}{\partial \widehat{\Theta}_{\mathcal{A}\psi}} = -\frac{\partial \Lambda_\psi}{\partial \widehat{\Theta}_{\mathcal{C}\psi}} = 2(\widehat{\Theta}_{\mathcal{A}\psi} - \widehat{\Theta}_{\mathcal{C}\psi})$, the update laws (30) are derived using the gradient descent method.

Taking the time derivative of $\Lambda_\psi(t)$ yields

$$\dot{\Lambda}_\psi(t) = \mathrm{Tr}\Big(\frac{\partial \Lambda_\psi(t)}{\partial \widehat{\Theta}_{\mathcal{A}\psi}(t)}\dot{\widehat{\Theta}}_{\mathcal{A}\psi}(t) + \frac{\partial \Theta_\psi(t)}{\partial \widehat{\Theta}_{\mathcal{C}\psi}(t)}\dot{\widehat{\Theta}}_{\mathcal{C}\psi}(t)\Big) = -\frac{\lambda_{\mathcal{A}\psi}}{2} \times \mathrm{Tr}\Big(\frac{\partial \Lambda_\psi(t)}{\partial \widehat{\Theta}_{\mathcal{A}\psi}(t)}\big(\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi) + \mu_\Delta \mathcal{I}_\psi\big)\frac{\partial \Lambda_\psi(t)}{\partial \widehat{\Theta}_{\mathcal{A}\psi}(t)}\Big) \le 0.$$

Therefore, this inequality ensures that $\Lambda_\psi(t) \to 0$ by utilizing the update laws (30).

In this subsection, all closed-loop signals are semi-globally uniformly ultimately bounded via Lyapunov theory, allowing the derivation of the main control results.

Theorem 1: Consider the n-link robotic manipulator system described in (1). Under the virtual control law (23), the FTC law (29), and the NN weight update laws (20), (22), (24), and (30), the closed-loop system guarantees that all system signals remain bounded within a predefined time and that the tracking error evolves inside a prescribed-performance region.

Proof: To analyze the closed-loop system, consider the following nonnegative Lyapunov function candidate: $\mathbb{V}(t) = \frac{1}{2}\sum_{\psi=1}^2 \wp_\psi^2 + \frac{1}{2}\tilde{\phi}_\iota^2 + \frac{1}{2}\sum_{\psi=1}^2\Big(\widetilde{\Theta}^T_{\mathcal{U}\psi}\widetilde{\Theta}_{\mathcal{U}\psi} + \widetilde{\Theta}^T_{\mathcal{C}\psi}\widetilde{\Theta}_{\mathcal{C}\psi} + \widetilde{\Theta}^T_{\mathcal{A}\psi}\widetilde{\Theta}_{\mathcal{A}\psi}\Big)$, where $\tilde{\phi}_\iota = \hat{\phi}_\iota - \phi^*_\iota$ denotes the estimation of the unknown bounded additive fault, and $\widetilde{\Theta}_{\mathcal{U}\psi} = \widehat{\Theta}_{\mathcal{U}\psi} - \Theta^*_{\mathcal{U}\psi}$, $\widetilde{\Theta}_{\mathcal{C}\psi} = \widehat{\Theta}_{\mathcal{C}\psi} - \Theta^*_{\mathcal{C}\psi}$, $\widetilde{\Theta}_{\mathcal{A}\psi} = \widehat{\Theta}_{\mathcal{A}\psi} - \Theta^*_{\mathcal{A}\psi}$, denote the identifier, critic, and actor NN weight estimation errors, respectively. Taking the time derivative of $\mathbb{V}(t)$ using (10) and (25) yields

$$\dot{\mathbb{V}}(t) = \wp_1\big(\varkappa_\sigma\eta_\varsigma(\wp_2 + \varphi_1 - \dot{\theta}_d - \hbar_{d1})\big) + \wp_2\big(\mathcal{M}_0^{-1}(\xi_{\beta\bar{\zeta}_\beta} + \gamma_\beta + d) - \ddot{q}_d + \ell - \dot{\varphi}\big) + \tilde{\phi}_\iota\dot{\hat{\phi}}_\iota + \sum_{\psi=1}^2\Big(\widetilde{\Theta}^T_{\mathcal{U}\psi}\dot{\widehat{\Theta}}_{\mathcal{U}\psi} + \widetilde{\Theta}^T_{\mathcal{C}\psi}\dot{\widehat{\Theta}}_{\mathcal{C}\psi} + \widetilde{\Theta}^T_{\mathcal{A}\psi}\dot{\widehat{\Theta}}_{\mathcal{A}\psi}\Big).$$

Applying (16) and (29), it can be derived that

$$\dot{\mathbb{V}}(t) = -\varrho_{a1}\wp_1^2 + \varkappa_\sigma\eta_\varsigma\wp_1\big(\wp_2 - \dot{\theta}_d - \hbar_{d1}\big) - \frac{\wp_1}{2}\widehat{\Theta}^T_{\mathcal{A}_1}\mathcal{Q}_{\Im_1}(\bar{\varkappa}_1) - \frac{\wp_1}{2}\widehat{\Theta}^T_{\mathcal{U}_1}\mathcal{Q}_{\mathcal{U}_1}(\bar{\varkappa}_1) - \varrho_{a2}\wp_2^2 + \mathcal{M}_0^{-1}\wp_2\gamma_\beta + \mathcal{M}_0^{-1}\wp_2 d + \wp_2 \times\big(-\ddot{q}_d + \ell - \dot{\varphi}_1\big) - \frac{\wp_2}{2}\widehat{\Theta}^T_{\mathcal{A}_2}\mathcal{Q}_{\Im_2}(\bar{\varkappa}_2) - \frac{\wp_2}{2}\widehat{\Theta}^T_{\mathcal{U}_2}\mathcal{Q}_{\mathcal{U}_2}(\bar{\varkappa}_2) + \tilde{\phi}_\iota\big(\|\wp_2\|_1 - \lambda_\phi\hat{\phi}_\iota\big) + \sum_{\psi=1}^2\big(\widetilde{\Theta}^T_{\mathcal{U}\psi}\dot{\widehat{\Theta}}_{\mathcal{U}\psi} + \widetilde{\Theta}^T_{\mathcal{C}\psi}\dot{\widehat{\Theta}}_{\mathcal{C}\psi} + \widetilde{\Theta}^T_{\mathcal{A}\psi}\dot{\widehat{\Theta}}_{\mathcal{A}\psi}\big).$$

Consequently, we have

$$\dot{\mathbb{V}}(t) = \sum_{\psi=1}^2\Big(-\varrho_{a\psi}\wp_\psi^2 + \wp_\psi\big(-\frac{1}{2}\widehat{\Theta}^T_{\mathcal{A}_\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi) - \widehat{\Theta}^T_{\mathcal{U}_\psi}\mathcal{Q}_{\mathcal{U}_\psi}\big)\Big) + \varkappa_\sigma\eta_\varsigma\wp_1\big(\wp_2 - \dot{\theta}_d - \hbar_{d1}\big) + \wp_2\big(\mathcal{M}_0^{-1}(\gamma_\beta + d) - \ddot{q}_d + \ell - \dot{\varphi}_1\big) + \tilde{\phi}_\iota\big(\|\wp_2\|_1 - \lambda_\phi\hat{\phi}_\iota\big) + \sum_{\psi=1}^2\big(\widetilde{\Theta}^T_{\mathcal{U}\psi}\dot{\widehat{\Theta}}_{\mathcal{U}\psi} + \widetilde{\Theta}^T_{\mathcal{C}\psi}\dot{\widehat{\Theta}}_{\mathcal{C}\psi} + \widetilde{\Theta}^T_{\mathcal{A}\psi}\dot{\widehat{\Theta}}_{\mathcal{A}\psi}\big).$$

Next, applying Young's inequality to handle cross-terms gives

$$\begin{cases}\varkappa_\sigma\eta_\varsigma\wp_1\wp_2 \le \frac{\varkappa_\sigma^2\eta_\varsigma^2}{2}\wp_1^2 + \frac{1}{2}\wp_2^2, \mathcal{M}_0^{-1}\wp_2 d \le \frac{1}{2\mathcal{M}_0^2}\wp_2^2 + \frac{1}{2}\bar{d}^2,\\ -\frac{1}{2}\widehat{\Theta}^T_{\mathcal{A}_\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi) \le \frac{1}{4}\wp_\psi^2 + \frac{1}{4}\widehat{\Theta}^T_{\mathcal{A}_\psi}\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widehat{\Theta}_{\mathcal{A}_\psi}.\end{cases} \tag{31}$$

Substituting these inequalities yields

$$\dot{\mathbb{V}}(t) = \sum_{\psi=1}^2\Big(-\varrho_{a\psi}\wp_\psi^2 - \frac{1}{2}\wp_\psi^2 + \wp_\psi\big(-\widehat{\Theta}^T_{\mathcal{U}_\psi}\mathcal{Q}_{\mathcal{U}_\psi}(\bar{\varkappa}_\psi) + \mathcal{U}_\psi(\bar{\varkappa}_\psi)\big) + \frac{1}{4}\widehat{\Theta}^T_{\mathcal{A}_\psi}\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widehat{\Theta}_{\mathcal{A}_\psi} + \widetilde{\Theta}^T_{\mathcal{U}\psi}\big(-\lambda_{\Delta\varepsilon}\widehat{\Theta}_{\mathcal{U}_1} - \mathcal{Q}_{\mathcal{U}_\psi}(\bar{\varkappa}_\psi) \times \wp_\psi\mu_\psi\big) + \widetilde{\Theta}^T_{\mathcal{C}\psi}\big(\lambda_{\mathcal{C}\psi}\big(\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi) + \mu_\Delta\mathcal{I}_m\big)\widehat{\Theta}_{\mathcal{C}\psi}\big) + \widetilde{\Theta}^T_{\mathcal{A}\psi} \times\big(-\big(\lambda_{\mathcal{A}\psi}\widehat{\Theta}_{\mathcal{C}\psi} + \lambda_{\mathcal{A}\psi}(\widehat{\Theta}_{\mathcal{A}\psi} - \widehat{\Theta}_{\mathcal{C}\psi})\big)\big(\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi) + \mu_\Delta\mathcal{I}_m\big) - \lambda_\phi\tilde{\phi}_\iota\hat{\phi}_\iota + \frac{1}{2}\bar{d}^2. \tag{32}$$

From the definitions $\tilde{\phi}_\iota = \hat{\phi}_\iota - \phi^*_\iota$, $\widetilde{\Theta}_{\mathcal{U}\psi} = \widehat{\Theta}_{\mathcal{U}\psi} - \Theta^*_{\mathcal{U}\psi}$, $\widetilde{\Theta}_{\mathcal{C}\psi} = \widehat{\Theta}_{\mathcal{C}\psi} - \Theta^*_{\mathcal{C}\psi}$, $\widetilde{\Theta}_{\mathcal{A}\psi} = \widehat{\Theta}_{\mathcal{A}\psi} - \Theta^*_{\mathcal{A}\psi}$, the following relations can be obtained:

$$\begin{cases}\tilde{\phi}_\iota\hat{\phi}_\iota = \frac{1}{2}\tilde{\phi}_\iota^T\tilde{\phi}_\iota + \frac{1}{2}\hat{\phi}_\iota^T\hat{\phi}_\iota - \frac{1}{2}\phi_\iota^{*T}\phi^*_\iota,\\ \widetilde{\Theta}^T_{\mathcal{U}\psi}\widehat{\Theta}_{\mathcal{U}\psi} = \frac{1}{2}\widetilde{\Theta}^T_{\mathcal{U}\psi}\widetilde{\Theta}_{\mathcal{U}\psi} + \frac{1}{2}\widehat{\Theta}^T_{\mathcal{U}\psi}\widehat{\Theta}_{\mathcal{U}\psi} - \frac{1}{2}\Theta^{*T}_{\mathcal{U}\psi}\Theta^*_{\mathcal{U}\psi},\\ \widetilde{\Theta}^T_{\mathcal{C}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widehat{\Theta}_{\mathcal{C}\psi} \le \frac{1}{2}\widetilde{\Theta}^T_{\mathcal{C}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widetilde{\Theta}_{\mathcal{C}\psi} + \frac{1}{2}\widehat{\Theta}^T_{\mathcal{C}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widehat{\Theta}_{\mathcal{C}\psi} - \frac{1}{2}\Theta^{*T}_{\mathcal{C}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}\Theta^*_{\mathcal{C}\psi},\\ \widetilde{\Theta}^T_{\mathcal{A}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widehat{\Theta}_{\mathcal{C}\psi} \le \frac{1}{2}\widetilde{\Theta}^T_{\mathcal{A}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widetilde{\Theta}_{\mathcal{A}\psi} + \frac{1}{2}\widehat{\Theta}^T_{\mathcal{A}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widehat{\Theta}_{\mathcal{A}\psi} - \frac{1}{2}\Theta^{*T}_{\mathcal{A}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}\Theta^*_{\mathcal{A}\psi}\end{cases} \tag{33}$$

By Young's inequality, one has

$$\begin{cases}-(\lambda_{\mathcal{C}\psi} - \lambda_{\mathcal{A}\psi})\widetilde{\Theta}^T_{\mathcal{A}\psi}\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widehat{\Theta}_{\mathcal{C}\psi} \le \frac{\lambda_{\mathcal{A}\psi} - \lambda_{\mathcal{C}\psi}}{2}\widetilde{\Theta}^T_{\mathcal{A}\psi} \times\mathcal{Q}_{\Im_\psi}(\bar{\varkappa}_\psi)\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widetilde{\Theta}_{\mathcal{A}\psi} + \frac{\lambda_{\mathcal{A}\psi} - \lambda_{\mathcal{C}\psi}}{2}\widetilde{\Theta}^T_{\mathcal{C}\psi}\mathcal{Q}_{\Im_\psi}\mathcal{Q}^T_{\Im_\psi}(\bar{\varkappa}_\psi)\widetilde{\Theta}_{\mathcal{C}\psi},\\ \wp_\psi\kappa_{\mathcal{U}\psi} \le \frac{1}{2}\wp_\psi^2 + \frac{1}{2}\bar{\kappa}_{\mathcal{U}\psi},\end{cases} \tag{34}$$

Substituting (33) and (34) into (32) yields

$$\dot{\mathbb{V}}(t) \le \sum_{\psi=1}^2\Big(-(\varrho_{a\psi} - \frac{1}{2})\wp_\psi^2 - \frac{1}{2}\lambda_{\Delta\psi}\widetilde{\Theta}^T_{\mathcal{U}\psi}\widetilde{\Theta}_{\mathcal{U}\psi} - \frac{1}{2}\lambda_{\Delta\psi}\widetilde{\Theta}^T_{\mathcal{U}\psi} \times\widetilde{\Theta}_{\mathcal{U}\psi} - \frac{\lambda_{\mathcal{A}\psi}}{2}\widetilde{\Theta}^T_{\mathcal{A}\psi}\big(\mathcal{Q}_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\mathcal{Q}^T_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon) + \mu_\Delta\mathcal{I}_m\big)\widetilde{\Theta}_{\mathcal{A}\psi} - \frac{\lambda_{\mathcal{C}\psi}}{2} \times\widetilde{\Theta}^T_{\mathcal{C}\psi}\big(\mathcal{Q}_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\mathcal{Q}^T_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon) + \mu_\Delta\mathcal{I}_m\big)\widetilde{\Theta}_{\mathcal{C}\psi} + \big(\frac{\lambda_{\mathcal{A}\psi}}{2} - \frac{1}{4}\big)\widehat{\Theta}^T_{\mathcal{A}\psi} \times\mathcal{Q}_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\mathcal{Q}^T_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\widehat{\Theta}_{\mathcal{A}\psi} + \big(\frac{\lambda_{\mathcal{A}\psi}}{2} - \lambda_{\mathcal{C}\psi}\big)\widehat{\Theta}^T_{\mathcal{C}\psi}\mathcal{Q}_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\mathcal{Q}^T_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon) \times\widehat{\Theta}_{\mathcal{C}\psi} - \frac{\lambda_{\mathcal{A}\psi}}{2}\widehat{\Theta}_{\mathcal{A}\psi}\mu_\Delta\mathcal{I}_m\widehat{\Theta}_{\mathcal{A}\psi}\Big) - \frac{\lambda_\phi}{2}(\tilde{\phi}_\iota^2 + \hat{\phi}_\iota^2) + W(t), \tag{35}$$

where $W(t) = \frac{1}{2}\bar{d}^2 + \frac{\lambda_\phi}{2}\|\phi_\iota\|^2 + \sum_{\psi=1}^2\big(\frac{\lambda_{\Delta\psi}}{2}\|\Theta_{\mathcal{U}\psi}\|^2 + \frac{(\lambda_{\mathcal{A}\psi} + \lambda_{\mathcal{C}\psi})}{2}\|\Theta^{*T}_{\Im\psi}\mathcal{Q}_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\|^2$, which is bounded by a positive constant ϖ, i.e.,$|W(t)| \le \varpi$. Under the design conditions $\varrho_{a\psi} > \frac{1}{2}$, $\lambda_{\mathcal{C}\psi} > \frac{\lambda_{\mathcal{A}\psi}}{2}$, and $\lambda_{\mathcal{A}\psi} > \frac{1}{2}$, it follows from (35) that

$$\dot{\mathbb{V}} \le -\sum_{\psi=1}^2(\varrho_{a\psi} - \frac{1}{2})\wp_\psi 2 - \frac{\lambda_\phi}{2}\tilde{\phi}_\iota^2 + \sum_{\psi=1}^2\Big(-\frac{\lambda_{\Delta\psi}}{2}\widetilde{\Theta}^T_{\mathcal{U}\psi}\widetilde{\Theta}_{\mathcal{U}\psi} - \frac{\lambda_{\mathcal{C}\psi}}{2}\widetilde{\Theta}^T_{\mathcal{A}\psi}\big(\mathcal{Q}_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\mathcal{Q}^T_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon) + \mu_\Delta\mathcal{I}_m\big)\widetilde{\Theta}_{\mathcal{A}\psi}$$

$$
\begin{aligned}
&- \frac{\lambda_{\mathcal{C}\psi}}{2}\widetilde{\Theta}_{\mathcal{C}\psi}^{T}\big(\mathcal{Q}_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\mathcal{Q}_{\Im_\varepsilon}^{T}(\bar{\varkappa}_\varepsilon)+\mu_\Delta\mathcal{I}_m\big)\widetilde{\Theta}_{\mathcal{C}\psi}\Big)+\varpi \\
&\le -\sum_{\psi=1}^{2}\big(\frac{2\varrho_{a\psi}-1}{2}\big)\wp_\psi^2-\frac{\lambda_\phi}{2}\tilde{\phi}_\iota^2+\sum_{\psi=1}^{2}\Big(-\frac{\lambda_{\Delta\psi}}{2}\widetilde{\Theta}_{\mathcal{U}\psi}^{T}\widetilde{\Theta}_{\mathcal{U}\psi} \\
&- \frac{\lambda_{\mathcal{C}\psi}}{2}\lambda_{\mathcal{Q}\Im_\psi}^{\min}\widetilde{\Theta}_{\mathcal{C}\psi}^{T}\widetilde{\Theta}_{\mathcal{C}\psi}-\frac{\lambda_{\mathcal{C}\psi}}{2}\lambda_{\mathcal{Q}\Im_\psi}^{\min}\widetilde{\Theta}_{\mathcal{A}\psi}^{T}\widetilde{\Theta}_{\mathcal{A}\psi}\Big)+\varpi. \quad (36)
\end{aligned}
$$

Let $\xi_a = \min\{(2\varrho_{a\psi}-1), \lambda_\phi, \lambda_{\mathcal{C}\psi}, \lambda_{\mathcal{C}\psi}\lambda_{\mathcal{Q}\Im_\psi}^{\min}\}$, where $\lambda_{\mathcal{Q}\Im_\psi}^{\min}$ denotes the minimum eigenvalue of $\mathcal{Q}_{\Im_\varepsilon}(\bar{\varkappa}_\varepsilon)\mathcal{Q}_{\Im_\varepsilon}^{T}(\bar{\varkappa}_\varepsilon)$. From inequality (36), the time derivative of the Lyapunov function satisfies $\dot{V}(t) \le -\xi_a V(t)+\varpi$. Solving this differential inequality yields $V(t) \le e^{-\xi_a t}V(0)+\frac{\varpi}{\xi_a}$. Taking the limit as $t\to\infty$, one obtains $\limsup_{t\to\infty} V(t) \le \frac{\varpi}{\xi_a}$. Therefore, for any given $\varepsilon>0$, there exists a finite time $\mathcal{T}_d$ such that $V(t)\le \frac{\varpi}{\xi_a}+\bar{k}, \forall t\ge\mathcal{T}_d$. Moreover, there exists a positive constant $\xi_b>0$ satisfying $V(t)\ge\xi_b\|\wp_\psi(t)\|^2$. Then $\|\wp_\psi(t)\|\le\sqrt{\frac{\varpi}{\xi_a}+\frac{\bar{k}}{\xi_b}}, \forall t\ge\mathcal{T}_d$. Hence, all composite error signals $\wp_\psi$ are semi-globally uniformly ultimately bounded. Moreover, by following similar analytical steps, the estimation errors of the adaptive parameters $\tilde{\phi}_\iota, \widetilde{\Theta}_{\mathcal{U}\psi}, \widetilde{\Theta}_{\mathcal{C}\psi}$, and $\widetilde{\Theta}_{\mathcal{A}\psi}$ are also ultimately bounded.

Remark 4: For robotic manipulator joint tracking in the presence of unknown dynamics, external disturbances, and actuator faults, the design parameters $\varkappa_a, \varkappa_b, \varkappa_c, \varkappa_\sigma, \eta_\varsigma, \mathcal{T}_d, \mu_\Delta, \lambda_{\Delta\psi}, \lambda_{A\psi}$, and $\lambda_{C\psi}$ collectively determine both the transient and steady-state performance of the closed-loop system. The prescribed time $\mathcal{T}_d$ and the steady-state constraint parameter μ_Δ explicitly define the desired convergence speed and tracking accuracy, where a smaller $\mathcal{T}_d$ enforces faster error convergence and a smaller μ_Δ results in tighter steady-state error bounds. The control gain $\varkappa_c$ primarily scales the control effort and accelerates error decay without affecting the final tracking accuracy. The weighting factor $\varkappa_\sigma$ balances position and velocity tracking errors in the filtered error formulation and is typically selected as $\varkappa_\sigma = 1$ when no priority is imposed. The learning-related parameters $\varkappa_a$, $\varkappa_b$, η_ς, $\lambda_{\Delta\psi}$, $\lambda_{\mathcal{A}\psi}$, and $\lambda_{\mathcal{C}\psi}$ govern the adaptation rates and convergence behavior of the actor–critic–identifier networks, influencing learning speed and fault estimation dynamics. Consequently, parameter tuning should comprehensively balance control accuracy, convergence rate, and transient performance.

Remark 5: In recent years, significant progress has been made in adaptive and robust FTC for robotic manipulators, including prescribed-performance and fixed-time tracking controllers [6], [9], [18], [22], [25]. However, many approaches assume ideal actuators or consider only matched disturbances, limiting applicability under composite actuator faults such as PLOE and TLOE. Compared with robust adaptive flexible prescribed-performance control [3], performance tracking control [20], and adaptive NN prescribed-performance controllers [23], our method explicitly handles composite actuator faults and ensures tracking errors converge within practical prescribed-time bounds. RL-based fixed-time trajectory tracking controllers [26] perform well under normal conditions but do not address actuator faults or prescribed-performance constraints. Similarly, adaptive fault-tolerant RL controllers [32], [33] show strong adaptability but limited fault-tolerance under composite faults. Our work bridges this gap by combining fault tolerance, prescribed-time convergence, and reduced computational effort, offering a comprehensive solution for practical robotic manipulator applications.

IV. Simulation Results

In this section, the validity of the developed controller is evaluated, and Algorithm 1 outlines the steps of the controller design procedure.

A. Simulation Verification: A 2-DOF robotic manipulator is employed for the simulations, and its schematic and physical parameters are presented in Fig. 2 and Table II, respectively. The manipulator dynamics follow the formulation described below:

$$
\mathcal{M}(\theta)=\begin{bmatrix} r_1+m_2l_1l_2\cos\theta_2 & r_2+J_2+\frac{1}{2}m_2l_1l_2\cos\theta_2 \\ r_2+J_2+\frac{1}{2}m_2l_1l_2\cos\theta_2 & r_2+J_2 \end{bmatrix}
$$

Algorithm 1 PPC-based Fault-Tolerant Tracking Control of Robotic Manipulators via Adaptive NN-based RL

Require: Desired joint trajectory $\theta_d(t), \dot{\theta}_d(t)$; control parameters $\varkappa_a, \varkappa_b, \varkappa_c, \varkappa_\sigma, \eta_\varsigma, \mathcal{T}_d, \mu_\Delta, \lambda_{\Delta\psi}, \lambda_{\mathcal{A}\psi}, \lambda_{\mathcal{C}\psi}$.

Ensure: Estimated parameter vectors $\widehat{\Theta}_{\mathcal{U}\psi}$, $\widehat{\Theta}_{\mathcal{C}\psi}$, $\widehat{\Theta}_{\mathcal{A}\psi}$, uncertainty compensation $\hat{\phi}_\iota$, and control torque τ.

1: Move the robotic manipulator to the prescribed initial configuration.
2: Initialize NN weights $\widehat{\Theta}_{\mathcal{U}\psi}(0)$, $\widehat{\Theta}_{\mathcal{C}\psi}(0)$, $\widehat{\Theta}_{\mathcal{A}\psi}(0)$, $\hat{\phi}_\iota(0)$, and control input $\tau(0)$.
3: **for** each time instant t **do**
4: Measure the joint position $\theta(t)$ and velocities $\dot{\theta}(t)$.
5: Compute the tracking errors $\wp_1$ and $\wp_2$ using (10) and (25).
6: Compute the virtual control input $\hat{\phi}_1$ according to (23).
7: Update the identifier, critic, and actor NN weight vectors $\widehat{\Theta}_{\mathcal{U}\psi}$, $\widehat{\Theta}_{\mathcal{C}\psi}$, and $\widehat{\Theta}_{\mathcal{A}\psi}$ using the adaptive laws in (30).
8: Estimate the model uncertainty compensation term $\hat{\phi}_\iota$ via (30).
9: Compute the fault-tolerant joint control input $\hat{\vartheta}$ using (29).
10: Generate and apply the control torque command τ to the robotic actuators according to (1).
11: **end for**

$$
\mathcal{C}(\theta,\dot{\theta})=\begin{bmatrix} -\frac{1}{2}m_2l_1l_2\dot{\theta}_2\sin\theta_2 & -\frac{1}{2}m_2l_1l_2(\dot{\theta}_1+\dot{\theta}_2)\sin\theta_2 \\ \frac{1}{2}m_2l_1l_2\dot{\theta}_1\sin\theta_2 & 0 \end{bmatrix}
$$

$$
\mathcal{G}(\theta)=\begin{bmatrix} \frac{1}{2}m_1gl_2\cos\theta_1+m_2gl_1\cos\theta_1+\frac{1}{2}m_2gl_2\cos(\theta_1+\theta_2) \\ \frac{1}{2}m_2gl_2\cos(\theta_1+\theta_2) \end{bmatrix}
$$

where $r_1 = \frac{1}{4}m_1l_1^2+r_2+m_2l_1^2+J_1+J_2$, and $r_2=\frac{1}{4}m_2l_2^2$. The physical parameters of the 2-DOF robotic manipulator used in this study, consistent with [26], are summarized in Table II. The manipulator is subjected to disturbance torque as $w(t) = [0.7(\sin(0.1t)+1,\ 0.7(\cos(0.1t)+1]^T$N.m.

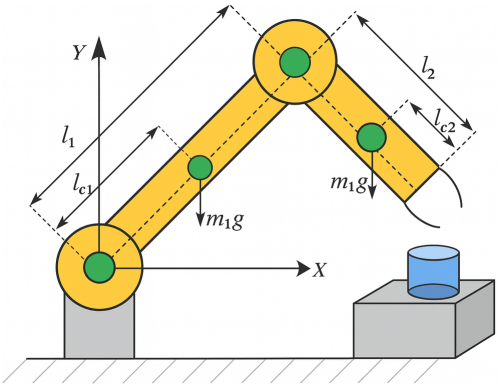

Fig. 2. Diagram of the 2-DOF robot manipulator.

TABLE II
PHYSICAL PARAMETERS OF THE 2-DOF ROBOT MANIPULATOR

Parameters	Descriptions	Values
m_1	Mass of joint 1	1.8 kg
m_2	Mass of joint 2	1.6 kg
J_1	Moment of joint 1	0.6 kg $\cdot$ m^2
J_2	Moment of joint 2	0.8 kg $\cdot$ m^2
L_1	Length of joint 1	1.7 m
L_2	Length of joint 2	1.5 m
$\bar{L}_1$	Mass center of joint 1	1.2 m
$\bar{L}_2$	Mass center of joint 2	0.8 m
g	Acceleration of gravity	9.8 m/s^2

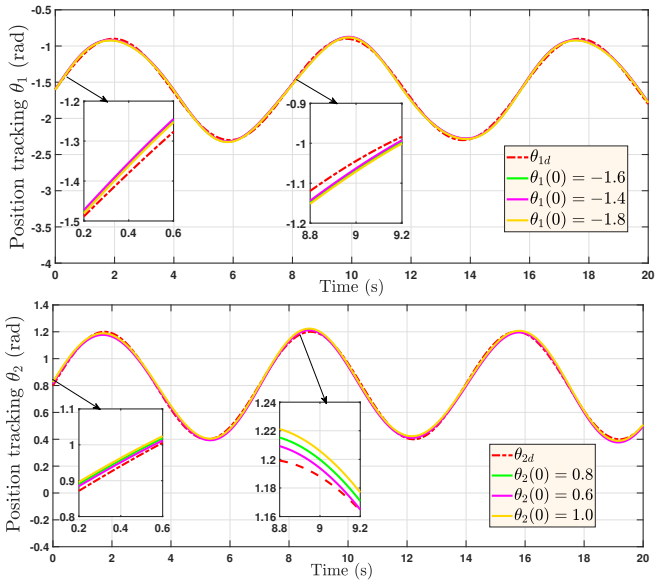

Fig. 3. Joint position tracking performance of joint 1 and joint 2 under different initial conditions.

TABLE III
GAUSSIAN BASIS FUNCTIONS AND RL NETWORK PARAMETERS

Parameter	Value
Identifier basis	$\mathcal{Q}_{\Im_1} = \exp\Big(-\frac{(\delta_1-7+\frac{14\varsigma}{21})^2}{b_1^2}\Big)$
	$\mathcal{Q}_{\Im_2} = \exp\Big(-\frac{\|[\delta_2-[7,7]^T+\frac{1}{2}[\varsigma,\varsigma]^T\|^2}{b_2^2}\Big)$
Critic-actor basis	$\mathcal{Q}_{\mathfrak{I}_1} = \exp\Big(-\frac{\|[\delta_1,\wp_1]^T-[7,7]^T+\frac{14}{21}[\varsigma,\varsigma]^T\|^2}{b_1^2}\Big)$
	$\mathcal{Q}_{\mathfrak{I}2} = \exp\Big(-\frac{\|[\delta_2,\wp_2]^T-[7,7]^T+\frac{1}{2}[\varsigma,\varsigma]^T\|^2}{b_2^2}\Big)$
Learning rates	$\lambda_{\Delta 1} = 0.76,\ \lambda_{\mathcal{C}1} = 2.6,\ \lambda_{\mathcal{A}1} = 2.9,\ \lambda_{\Delta 2} = 0.41$ $\lambda_{\mathcal{C}2} = 2.5,\ \lambda_{\mathcal{A}2} = 2.8$
NN weighs	$\widehat{\Theta}_{\mathfrak{I}1} = \widehat{\Theta}_{\mathcal{C}1} = \widehat{\Theta}_{\mathcal{A}1} = [0.01,\ldots,0.01] \in \mathbb{R}^{21\times 1}$
	$\widehat{\Theta}_{\mathfrak{I}2} = \widehat{\Theta}_{\mathcal{C}2} = \widehat{\Theta}_{\mathcal{A}2} = [0.01,\ldots,0.01] \in \mathbb{R}^{28\times 1}$

The desired joint trajectories are selected as $\theta_d = \begin{bmatrix}\theta_{1d}\\ \theta_{2d}\end{bmatrix} = \begin{bmatrix}0.08\cos(0.2t)\\ 0.08\cos(0.28t) - 0.1\sin(0.04t)\end{bmatrix}$ rad, with $t \in [0, \mathcal{T}_d]$ and $\mathcal{T}_d = 2s$. To emulate actuator failures during manipulator operation, the faulty torque input $\tau(t)$ is defined as follows. Joint 1 experiences an LOE fault, modeled by $\tau_1(t) = 0.7\,\vartheta_1(t),\ t \geq 1$ s indicating a 30% reduction in torque capability, and joint 2 undergoes a stuck fault represented by $\tau_2(t) = 2,\ t \geq 1.5$s, which captures a typical lock-in-place failure in robotic manipulators. To approximate the unknown nonlinear dynamics, Gaussian basis functions are employed. For joint 1, $\mathcal{Q}_{\Im_1}(\varkappa_1)$ and $\mathcal{Q}_{\mathfrak{I}_1}(\varkappa_1)$ are configured with 21 neurons, and for joint 2, $\mathcal{Q}_{\Im_2}(\varkappa_2)$ and $\mathcal{Q}_{\mathfrak{I}_2}(\varkappa_2)$ are configured with 28 neurons, with all neuron centers uniformly distributed over $[-7, 7]$, and the detailed configurations are summarized in Table III. To guarantee the prescribed performance of the tracking errors, the following performance boundary parameters are selected: $\varkappa_a = 0.8, \varkappa_b = 0.78, \varkappa_c = 0.5$ $\mu_\Delta = 2.7,\ \varrho_{a1} = 30,\ \varrho_{a2} = 20$.

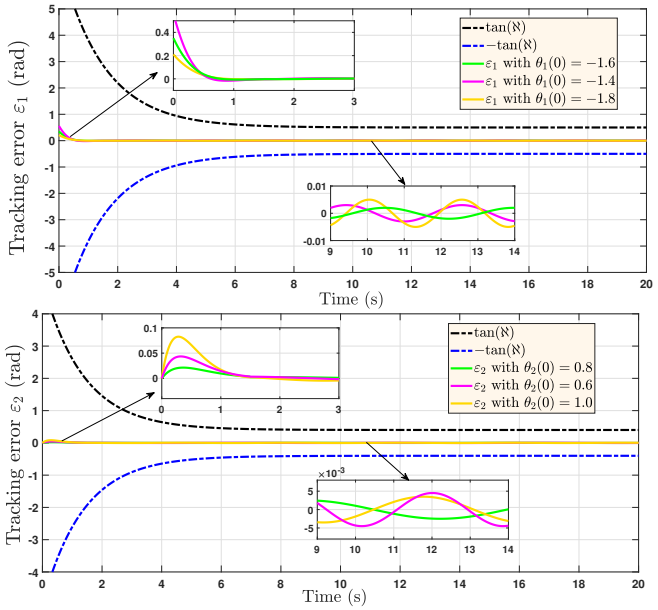

Fig. 4. Tracking errors of joint 1 and joint 2 under prescribed performance constraints for different initial conditions.

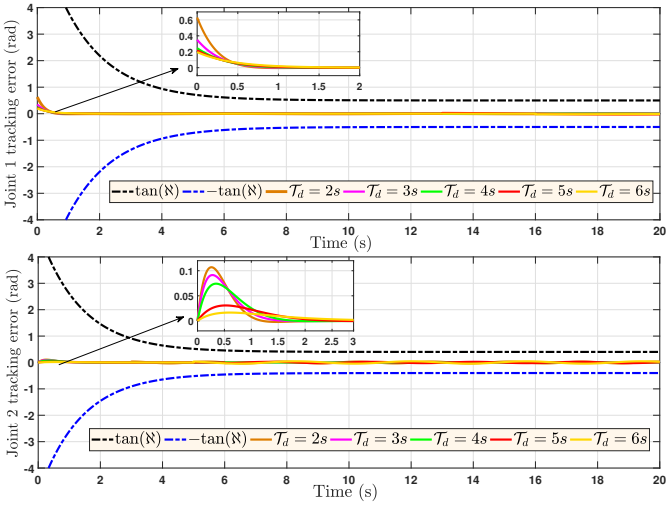

Fig. 5. Tracking errors of joint 1 and joint 2 under prescribed performance for different predefined times.

To validate the global control capability of the proposed design, the manipulator is tested under three different initial conditions with joint positions $\theta(0) \in \{[-1.6,\ 0.8]^T,\ [-1.4,\ 0.6]^T,\ [-1.8,\ 1.0]^T\}$. As shown in Fig. 3, the proposed control scheme ensures fast and accurate trajectory tracking for all initial conditions. Fig. 4 illustrates the transient joint tracking errors for different initial conditions, showing that larger deviations lead to higher transient peaks, smaller deviations produce smoother responses, and the controller effectively compensates for actuator faults, with errors converging near the origin. Fig. 5 shows the tracking errors of joint 1 and joint 2 over

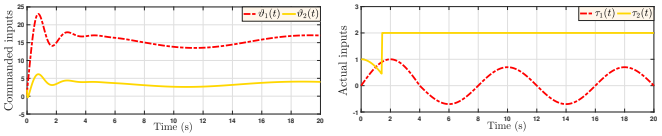

Fig. 6. Trajectories of commanded signals $\vartheta(t)$ and actual actuator torques $\tau(t)$.

TABLE IV
ACTUATOR TORQUE INPUTS FOR DIFFERENT FAULT SCENARIOS

Fault mode	$\tau_1(t)$	$\tau_2(t)$
fault 0	$0.5\vartheta_1(t),\ t \geq 1$	$2,\ t \geq 1.5$
fault 1	$0.2\vartheta_1(t),\ t \geq 1$	$2,\ t \geq 1.5$
fault 2	$0.2\vartheta_1(t),\ t \geq 1$	$-20,\ t \geq 1.5$
no fault	$\vartheta_1(t)$	$\vartheta_2(t)$

time for different desired convergence times, $\mathcal{T}_d = 2, 3, 4, 5, 6$ s, with the insets highlighting the first 2-3 seconds where $\mathcal{T}_d$ most affects convergence; all cases reach near-zero steady-state error, with smaller $\mathcal{T}_d$ (e.g., 2s) producing faster convergence but higher initial overshoot and larger $\mathcal{T}_d$ (e.g., 6s) reducing peak amplitude but slowing error decay, illustrating the trade-off between speed and transient response while maintaining robust steady-state tracking. Fig. 6 illustrates the trajectories of commanded signals $\vartheta(t)$ and actual actuator torques $\tau(t)$. In addition, Fig. 7 shows the tracking errors of both joints under no-fault, fault 0, fault 1, and fault 2, corresponding to the actuator conditions in Table IV. According to Table IV, actuator faults are implemented by modifying the joint torques after predefined instants. In fault 0, joint 1 has a 50% PLOE, and joint 2 is stuck at a constant torque. In fault 1, joint 1 suffers a 80% PLOE, and joint 2 remains stuck. In fault 2, joint 1 retains only 20% effectiveness, and joint 2 is locked at an extreme constant torque, representing a severe TLOE. In the no-fault case, both joints operate normally. As fault severity increases from fault 0 to fault 2, the peak tracking errors increase and convergence slows, but in all cases the errors remain bounded and converge within the predefined time, demonstrating strong robustness of the proposed controller against both PLOE and TLOE faults. Fig. 8 illustrates the behavior of actuator torques under various fault conditions. Fig. 9 confirms that the NN weight update laws $\widehat{\Theta}_{\mathcal{C}\psi}$, $\widehat{\Theta}_{\mathcal{A}\psi}(\psi = 1, 2)$ remain bounded and converge, and Fig. 10 shows that the cost functions $\bar{\chi}_1$ and $\bar{\chi}_2$ are reduced, indicating that the proposed algorithm effectively minimizes the cost function and enhances tracking performance.

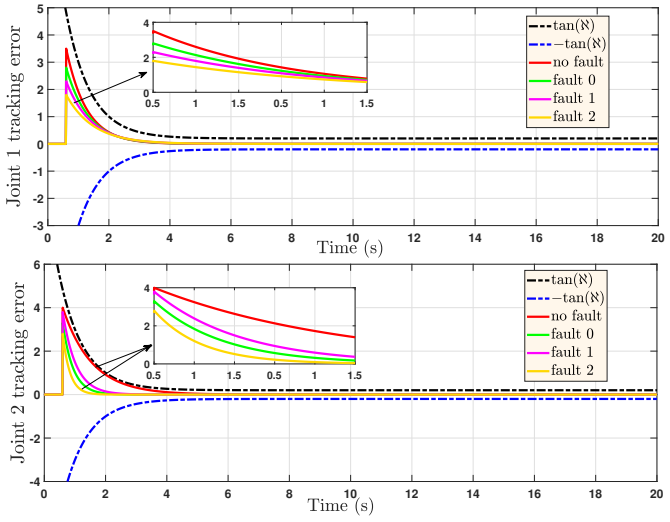

Fig. 7. Tracking errors of joint 1 and joint 2 with prescribed performance under different fault modes.

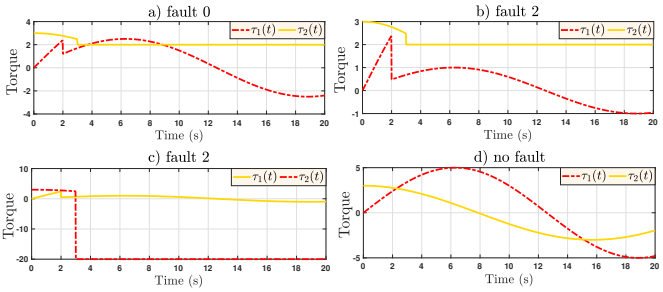

Fig. 8. Actuator torque inputs $\tau_1(t)$ and $\tau_2(t)$ under different fault scenarios: a) fault 0, b) fault 1, c) fault 2, and d) no fault.

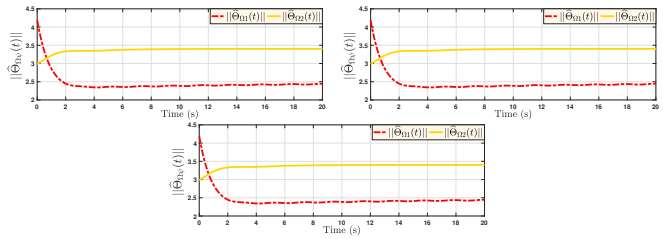

Fig. 9. Time evolution of the weight update norms for the identifier, critic, and actor learning networks.

B. Comparative Simulations: To demonstrate the effectiveness of the proposed RL-based prescribed-performance fault-tolerant control (RL-PPFTC), comparative simulations are conducted against three existing controllers under the initial condition $\theta(0) = [-1.3,\ 0.7]^T$. These controllers include: The benchmark controllers include: 1) prescribed performance-based FTC (PPFTC) [25]: An adaptive NN-based prescribed performance fault-tolerant control scheme with parameters $\varkappa_a = 2.1, \varkappa_b = 3.9, \varkappa_c = 11.5,\ \mu_\Delta = 7.1,\ \varrho_{a1} = 70,\ \varrho_{a2} = 40$; 2) PPC [23]: A finite-time predefined-time controller with $\varkappa_a = 0.8, \varkappa_b = 0.78, \varkappa_c = 0.5\ \mu_\Delta = 2.7,\ \varrho_{a1} = 30,\ \varrho_{a2} = 20$; 3) FTC [8]: An adaptive fixed-time fault-tolerant controller under composite actuator faults using the same parameter values as the proposed method. In contrast, the proposed control strategy achieves superior tracking performance, including faster transient response and higher steady-state accuracy, even in the presence of composite actuator faults, as shown in Fig. 11. Further, to evaluate control performance, tracking accuracy is quantified using several metrics: the overall tracking error (TE), root-mean-square error (RMSE), mean error (ME, $\bar{\varepsilon}$), standard deviation (SD), and maximum absolute error (MAE). The tracking error for each joint is defined as $TE = \|[\delta_1 - \theta_{1d}, \delta_2 - \theta_{2d}]^T\|$, where δ_ψ and $\theta_{\psi d}$ represent the actual and desired positions of the ψth joint, respectively. The corresponding performance metrics are calculated

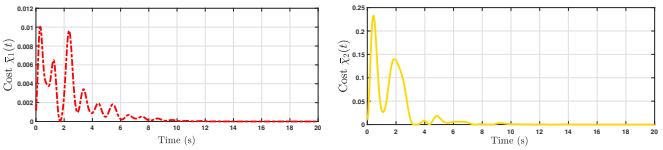

Fig. 10. Time evolution of the performance index.

as $RMSE = \sqrt{\frac{1}{m}\sum_{s=1}^{m} TE^2(s)}$, $\bar{\varepsilon} = \frac{1}{s}\sum_{s=1}^{m}\varepsilon_\psi(s)$, $SD = \sqrt{\frac{1}{m}\sum_{s=1}^{m}(\varepsilon_\psi(s)-\bar{\varepsilon})^2}$, $MAE = \max_{s=1,\ldots,m}|\varepsilon_\psi(s)|$, where m is the total number of samples and s is the sample index. These metrics jointly capture transient behavior, steady-state accuracy, and robustness to disturbances. The numerical results reported in Table V and Fig. 12 demonstrate that RL-PPFTC consistently outperforms all comparison methods, achieving reductions of approximately 81–85% in RMSE, 50–68% in ME, 85–88% in MAE, and more than 97% in SD. In contrast, PPFTC, PPC, and FTC show higher tracking errors and weaker robustness, particularly during dynamic operation. In addition, an RL-based neural controller reported in [26] (shown by the red curve in Fig. 7) is included for comparison. This controller does not explicitly address actuator failures nor enforce prescribed-performance constraints. As illustrated in Fig. 7, the red curve exhibits relatively large initial tracking errors and slower convergence, with the tracking error remaining comparatively high during the transient phase and, in fault scenarios, exceeding the prescribed performance bounds. Although in the no-fault case the actuator torques $\tau_1(t)$ and $\tau_2(t)$ follow their commanded inputs $\vartheta_1(t)$ and $\vartheta_2(t)$ reasonably well, noticeable tracking inaccuracies persist due to the lack of explicit performance guarantees. Under fault scenarios 0, 1, and 2, the actuator torques deviate significantly from their nominal values to reflect reduced or altered actuation capabilities, with fault 2 producing a pronounced negative torque in $\tau_2(t)$, indicating severe actuator malfunction. In contrast, the proposed control strategy maintains bounded tracking errors and achieves faster convergence, while ensuring that the control inputs satisfy the prescribed performance constraints governed by parameters $\varkappa_a, \varkappa_b, \varkappa_c$ as defined in (6). These results demonstrate that, unlike the RL controller in [26], the proposed method preserves robust tracking performance under both nominal and faulty operating conditions.

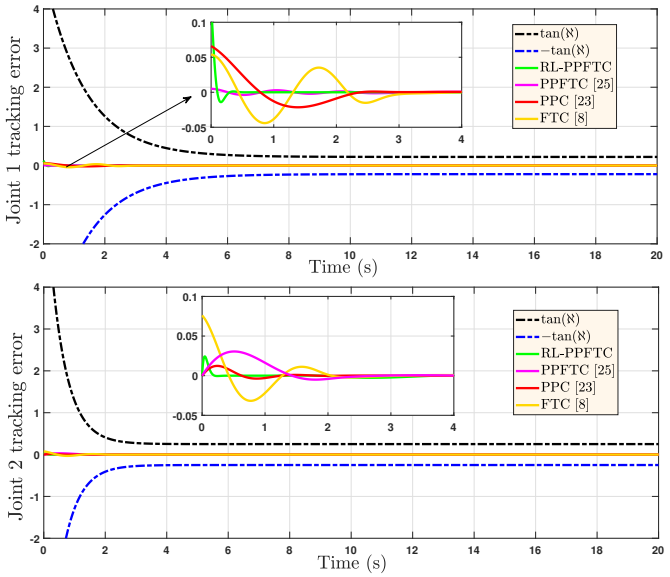

Fig. 11. Comparison of joint tracking errors for RL-PPFTC and existing controllers [8], [23], [25].

TABLE V
EVALUATION OF TRACKING PERFORMANCE FOR DIFFERENT CONTROL METHODS

Method	RMSE	ME	SD	MAE
RL-PPFTC	0.0113	0.0089	0.0015	0.0137
PPFTC [25]	0.0600	0.0200	0.0550	0.0900
PPC [23]	0.0750	0.0280	0.0650	0.1100
FTC [8]	0.0680	0.0180	0.0600	0.1000

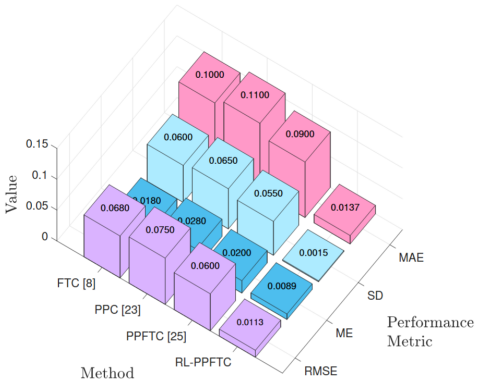

Fig. 12. Quantitative metrics of tracking performance for different control methods [8], [23], [25].

V. CONCLUSION

This paper proposed an RL-based prescribed-performance fault-tolerant tracking framework for robotic manipulators under composite actuator faults, including PLOE and TLOE. The control scheme employs a PPF with a filtered error transformation to achieve predefined-time tracking without initial condition restrictions. A robust control mechanism compensates for abrupt actuator faults and NN approximation errors, while the identifier-critic-actor RL architecture minimizes HJB approximation errors to optimize control performance. Simulation studies on a two-link manipulator demonstrate the method's superior tracking accuracy, fault tolerance, and robustness compared with conventional approaches. The results confirm that the proposed framework provides a practical and effective solution for high-precision control of robotic manipulators under uncertain dynamics and actuator faults.

REFERENCES

[1] S. Zheng, C. K. Ahn, M. Wan, Y. Xie, and P. Shi, "Adaptive cooperative output regulation for multiple flexible manipulators," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 54, pp. 4819–4831, 2024.

[2] G. I. Song, H. Y. Park, and J. H. Kim, "The H_∞ robust stability and performance conditions for uncertain robot manipulators," *IEEE/CAA J. Automatica Sinica*, vol. 12, pp. 270–272, 2025.

[3] X. Zhou, H. Wang, and Y. Tian, "Robust adaptive flexible prescribed performance tracking and vibration control for rigid-flexible coupled robotic systems with input quantization," *Nonlinear Dyn.*, vol. 112, pp. 1951–1969, 2024.

[4] M. X. Wang, S.L. Zhu, S. M. Liu, Y. Du, and Y. Q. Han, "Design of adaptive finite-time fault-tolerant controller for stochastic nonlinear systems with multiple faults," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, pp. 2492–2502, 2023.

[5] Q. Shen, P. Shi, and C. P. Lim, "Fuzzy adaptive fault-tolerant stability control against novel actuator faults and its application to mechanical systems," *IEEE Trans. Fuzzy Syst.*, vol. 32, pp. 2331–2340, 2024.

[6] L. Liu, L. Zhang, Y. Wang, and Y. Hou, "A novel robust fixed-time fault-tolerant tracking control of uncertain robot manipulators," *IET Control Theory Appl.*, vol. 15, pp. 195–208, 2021.

[7] B. Xiao and S. Yin, "An intelligent actuator fault reconstruction scheme for robotic manipulators," *IEEE Trans. Cybern.*, vol. 48, pp. 639–647, 2018.

[8] Y. Zhu, Z. Liu, B. Jiang, and Q. Zhu, "Adaptive fixed-time fuzzy fault-tolerant control for robotic manipulators with unknown friction and composite actuator faults," *J. Franklin Inst.*, vol. 361, pp. 107025, 2024.

[9] W. Liu, Z. Ye, D. Zhang, J. Cheng, and H. Yan, "Adaptive estimator-based nonsingular fast terminal sliding mode control of robotic manipulator systems under FDI attacks and actuator failure," *J. Franklin Inst.*, vol. 361, pp. 107129, 2024.

[10] Z. Anjum and Y. Guo, "Finite time fractional-order adaptive backstepping fault tolerant control of tobotic manipulator," *Int. J. Control Autom. Syst.*, vol. 19, pp. 301–310, 2021.

[11] M. Van, M. Mavrovouniotis, and S. S. Ge, "An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, pp. 1448–1458, 2019.

[12] Y. Hu, H. Yan, H. Zhang, M. Wang, and L. Zeng, "Robust adaptive fixed-time sliding-mode control for uncertain robotic systems with input saturation," *IEEE Trans. Cybern.*, vol. 53, pp. 2636–2646, 2023.

[13] S. Zhang, P. Yang, L. Kong, W. Chen, Q. Fu, and K. Peng, "Neural networks-based fault tolerant control of a robot via fast terminal sliding mode," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, pp. 4091–4101, 2021.

[14] M. Van and S. S. Ge, "Adaptive fuzzy integral sliding-mode control for robust fault-tolerant control of robot manipulators with disturbance observer," *IEEE Trans. Fuzzy Syst.*, vol. 29, pp. 1284–1296, 2021.

[15] J. Hu, X. Zhang, D. Zhang, Y. Chen, H. Ni, and H. Liang, "Finite-time adaptive super-twisting sliding mode control for autonomous robotic manipulators with actuator faults," *ISA Trans.*, vol. 144, pp. 342–351, 2024.

[16] Y. Zhu, W. Zhu, J. Liu, Q. G. Wang, and J. Yu, "Command-filtered finite-time fuzzy adaptive fault-tolerant control of output-constrained robotic manipulators with unknown dead-zones," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 70, pp. 2939–2943, 2023.

[17] M. Van, Y. Sun, S. McIlvanna, M.-N. Nguyen, M. O. Khyam, and D. Ceglarek, "Adaptive fuzzy fault-tolerant control for robot manipulators with fixed-time convergence," *IEEE Trans. Fuzzy Syst.*, vol. 31, pp. 3210–3219, 2023.

[18] Z. Anjum, Z. Sun, and B. Chen, "Disturbance-observer-based fault-tolerant control of robotic manipulator: A fixed-time adaptive approach," *IET Control Theory Appl.*, vol. 18, pp. 1398–1413, 2024.

[19] S. Sui, C. L. P. Chen, and S. Tong, "A novel adaptive NN prescribed performance control for stochastic nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, pp. 3196–3205, 2021.

[20] J. X. Zhang, K. D. Xu, and Q. G. Wang, "Prescribed performance tracking control of time-delay nonlinear systems with output constraints," *IEEE/CAA J. Autom. Sinica*, vol. 11, pp. 1557–1565, 2024.

[21] J. X. Zhang, J. G. Song, and P. Shi, "Decentralized robust global prescribed performance control of unknown interconnected nonlinear systems," *IEEE Trans. Ind. Electron.*, vol. 71, pp. 14701–14711, 2024.

[22] X. Zhou, H. Wang, and Y. Tian, "Robust adaptive flexible prescribed performance tracking and vibration control for rigid–flexible coupled robotic systems with input quantization," *Nonlinear Dyn.*, vol. 112, pp. 1951–1969, 2024.

[23] H. Ma, Q. Zhou, H. Li, and R. Lu, "Adaptive prescribed performance control of a flexible-joint robotic manipulator with dynamic uncertainties," *IEEE Trans. Cybern.*, vol. 52, pp. 12905–12915, 2022.

[24] W. Ding, J. X. Zhang, and P. Shi, "Adaptive fuzzy control of wheeled mobile robots with prescribed trajectory tracking performance," *IEEE Trans. Fuzzy Syst.*, vol. 32, pp. 4510–4521, 2024.

[25] J. Zhang, P. Han, Z. Wu, Q. Liu, and J. Yang "Nonfragile prescribed performance control of robot manipulators with actuator faults," *Int. J. Control Autom. Syst.*, vol. 22, pp. 3472–3481, 2024.

[26] S. Cao, L. Sun, J. Jiang, and Z. Zuo, "Reinforcement learning-based fixed-time trajectory tracking control for uncertain robotic manipulators with input saturation." *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, pp. 4584–4595, 2023.

[27] J. W. Zhu, Z. Y. Dong, Z. J. Yang, and X. Wang, "A new reinforcement learning fault-tolerant tracking control method with application to Baxter robot," *IEEE/ASME Trans. Mechatronics*, vol. 29, pp. 1331–1341, 2024.

[28] N. Wang, Y. Gao, C. Yang, and X. Zhang, "Reinforcement learning-based finite-time tracking control of an unknown unmanned surface vehicle with input constraints," *Neurocomputing*, vol. 484, pp. 26–37, 2022.

[29] S. Li, L. Ding, H. Gao, Y. J. Liu, N. Li, and Z. Deng, "Reinforcement learning neural network-based adaptive control for state and input time-delayed wheeled mobile robots," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 50, pp. 4171–4182, 2020.

[30] G. Narayanan, R. Karthikeyan, S. Lee, and S. Ahn, "Intelligent resilient security control for fractional-order multiagent networked systems using reinforcement learning and event-triggered communication mechanism," *IEEE Trans. Cybern.*, vol. 55, pp. 5103–5116, 2025.

[31] C. Yan, J. Xia, J. H. Park, and X. Xie, "Reinforcement learning-based adaptive event-triggered fuzzy control for cyclic switched stochastic nonlinear systems with actuator faults," *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 3, pp. 1131–1143, Mar. 2024.

[32] S. Roshanravan and S. Shamaghdari, "Adaptive fault-tolerant tracking control for affine nonlinear systems with unknown dynamics via reinforcement learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 1, pp. 569–580, Jan. 2024.

[33] H. Wang, S. Jia, S. Liu, X. Liu, "Adaptive finite-time optimal control for flexible-joint robots via an identifier-critic-actor reinforcement learning algorithm," *Nonlinear Dyn.*, vol. 113, pp. 32405–32426, 2025.