Prescribed-Time Fault-Tolerant Optimal Control of Continuum Surgical Robots via Composite Deep Learning

Recommended by Prof. T Parisini

Journal Pre-proof

Prescribed-Time Fault-Tolerant Optimal Control of Continuum Surgical Robots via Composite Deep Learning

Mohammad Jabari , Carmen Visconte , Giuseppe Quaglia , Med Amine Laribi

PII: S0947-3580(26)00067-1
DOI: https://doi.org/10.1016/j.ejcon.2026.101514
Reference: EJCON 101514

To appear in: *European Journal of Control*

Received date: 6 October 2025
Revised date: 1 April 2026
Accepted date: 26 April 2026

Please cite this article as: Mohammad Jabari , Carmen Visconte , Giuseppe Quaglia , Med Amine Laribi , Prescribed-Time Fault-Tolerant Optimal Control of Continuum Surgical Robots via Composite Deep Learning, *European Journal of Control* (2026), doi: https://doi.org/10.1016/j.ejcon.2026.101514

Highlights

- **Prescribed-time fault-tolerant control for surgical continuum robots**
- **Descriptor observer estimates sensor delays and faults jointly**
- **Composite DNN learning improves transient performance**
- **Compensator mitigates delay and saturation effects**
- **Outperforms a robust baseline in realistic scenarios**

Prescribed-Time Fault-Tolerant Optimal Control of Continuum Surgical Robots via Composite Deep Learning

M*ohammad Jabari* [a,b,*] *Carmen Visconte*[b] , *Giuseppe Quaglia* [b], *Med Amine Laribi* [c]

a Department of Informatics, Bioengineering, Robotics and Systems Engineering (DIBRIS), Università di Genova, Genova, Italy

b Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

c Department of GMSC, Pprime Institute, University of Poitiers, CNRS, ISEA-ENSMA, UPR 3346, Poitiers, France

∗Corresponding author Email addresses: mohammad.jabari@polito.it (Mohammad Jabari), carmen.visconte@polito.it (Carmen Visconte), giuseppe.quaglia@polito.it (Giuseppe Quaglia), med.amine.laribi@univ poitiers.fr (Med Amine Laribi)

Abstract:

Fault-tolerant control is essential for surgical continuum robots, where actuator and sensor faults, time-varying delays, input saturation, disturbances, and nonlinearities can rapidly degrade accuracy and safety. This paper proposes a prescribed-time fault-tolerant framework that guarantees convergence within a user-defined settling time, independent of initial conditions, under stated assumptions. A descriptor-form observer simultaneously estimates sensor delays and fault signals, while a deep neural network (DNN) approximates unmodeled dynamics. Adaptive laws compensate for disturbances and approximation errors, and a time-base generator enforces prescribed-time convergence of the estimation errors. Building on these estimates, a resilient near-optimal controller is synthesized, whose residual term is learned online by a DNN via composite learning that combines tracking and observer errors. An auxiliary compensator mitigates input-delay and saturation effects. Lyapunov analysis provides sufficient LMI conditions for prescribed-time closed-loop stabilization. Simulations on a tendon-driven continuum robot under simultaneous actuator and sensor faults, time-varying delays, and saturation demonstrate superior

tracking accuracy, faster convergence, and lower control effort compared with a representative robust baseline, highlighting the framework's strong potential for safe and precise surgical manipulation.

Keywords: Surgical continuum robots, fault-tolerant control, prescribed-time stabilization, descriptor-form observer, deep neural networks, composite learning, adaptive robust control

1. Introduction

Robotics has profoundly transformed human life. In surgery, systems such as the da Vinci robot have revolutionized minimally invasive procedures by enhancing precision while reducing patient discomfort, costs, and recovery times. However, these platforms require bulky operating-room setups and rigid instruments, limiting access to complex anatomies. To overcome these limitations, flexible continuum robots [1, 2] have emerged, enabling dexterous navigation and manipulation through narrow anatomical pathways with smooth, continuous bending and compliant interaction with soft tissue. Despite their clinical potential, continuum robots still face significant challenges in real surgical environments, including complex multi-actuator architectures and highly dynamic conditions [3, 4]. Existing models and control approaches often neglect one or more of these phenomena or treat them independently under idealized assumptions. The present work tackles the simultaneous presence of four major difficulties: (i) time-varying delays in both measurement and actuation channels, (ii) actuator input saturation, (iii) simultaneous sensor and actuator faults, and (iv) the requirement of convergence within a surgeon-specified time while preserving near-optimal performance. Addressing all four phenomena simultaneously in a unified fault-tolerant framework with proven prescribed-time convergence and online optimality has not yet been achieved in the literature. Key issues include measurement-channel delays, actuation-channel delays, sensor faults, actuator saturation, and nonlinear disturbances [5–8]. Such conditions can degrade performance or cause instability, with sensor faults posing direct risks to patient safety. Consequently, there is an urgent need for controllers that ensure stable operation under the simultaneous presence of faults, delays, and disturbances [9–13]. Numerous control strategies have been investigated, but each exhibits critical limitations when facing the combined challenges. Classical PID schemes [14] lack theoretical robustness guarantees under simultaneous faults and delays. Fuzzy-logic control methods [15] and model predictive control (MPC) [16–18] are sensitive to model uncertainty and impose high computational burden. Robust H∞ and sliding mode control improve disturbance rejection, yet conventional SMC achieves only finite-time stability dependent on initial conditions [3, 19]. While active fault-tolerant control (AFTC) and observer-based methods [20–24, 29] improve fault diagnosis and compensation, their integration with prescribed-time stability remains limited. Recent works have addressed only subsets of the problem, such as prescribed-performance fault-tolerant control [27] and prescribed-time neural control without fault-tolerant design [28], time-delay systems with saturation but without prescribed-time guarantees [30–33], or fuzzy-approximation-based FTC neglecting simultaneous input/output delays [34, 35]. None of them simultaneously tackles all these phenomena in a single unified framework.

To bridge this gap, this paper proposes a prescribed-time, deep learning–based active fault-tolerant control framework for tendon-driven continuum robots. The approach incorporates a prescribed-time observer for rapid and accurate estimation of sensor and actuator faults, time-varying delays, and saturation effects. Inspired by recent advances in learning-based control [36, 37], a deep neural network approximates online the residual of the optimal control law derived from the Hamilton–Jacobi–Bellman equation using composite learning that exploits both tracking error and observer estimation error. Computational complexity is drastically reduced compared with classical MPC or offline HJB solvers, since only a single critic-like DNN with analytic weight update rules is executed online. An auxiliary system dynamically compensates for actuator saturation, while novel control and observer terms ensure prescribed-time closed-loop stability. The framework is validated through simulations on a tendon-driven continuum robot subjected simultaneously to actuator/sensor faults, time-varying input/output delays, saturation, disturbances, and nonlinear uncertainties. Results demonstrate superior tracking accuracy, prescribed-time convergence, and reduced control effort compared with state-of-the-art methods [25, 26, 34, 35, 38].

The main contributions of this paper, which distinctly advance the state of the art, are as follows:

- A unified model and control design that simultaneously considers time-varying input and output delays, actuator saturation, sensor and actuator faults, and unmatched disturbances in continuum robots, unlike [30,35], which address only partial combinations.
- A novel prescribed-time observer and controller that guarantee convergence of both estimation and tracking errors within an arbitrary user-defined time, independent of initial conditions, going beyond the fixed-time results in [42,50] and existing prescribed-time delay-only approaches.
- Deep learning–based online approximation of the HJB optimal control law using composite (tracking + estimation) learning, achieving near-optimal performance while handling nonlinearities induced by faults and saturation.
- An adaptive auxiliary system for input saturation compensation integrated into the prescribed-time framework, preserving the prescribed-time convergence property (absent in [30–33]).
- A rigorous Lyapunov-based proof of prescribed-time stability for the overall closed-loop system under the simultaneous presence of all considered adverse phenomena, a result claimed here for the first time.

The remainder of this paper is organized as follows. Section 2 presents the mathematical modelling of the tendon-driven continuum robot. Section 3 introduces the mathematical foundations and the design procedure of the proposed prescribed-time fault-tolerant control framework. Section 4 presents the simulation results and comparative performance analysis. Section 5 concludes the paper and outlines future research directions.

2. Mathematical Modeling of the Continuum Robot

Continuum robots are highly adaptable mechanisms capable of smooth, continuous deformation, enabling precise and flexible motion in constrained environments [43]. A two-segment continuum robot with four degrees of freedom (DoF) is considered. Each segment is actuated by three inextensible cables routed through intermediate spacer disks. As shown in Fig. 1, for segment $i \in \{1,2\}$, the configuration variables are the bending magnitude θ_i and the bending-plane orientation δ_i. The segment lengths are defined by $l_1$and l_2 , and the total backbone length is $l = l_1 + l_2$. The parameters used in this study are listed in Table 1.

Table 1. Parameter values of the considered continuum robot.

Parameter	Designation	Value
l	Bending section length	$0.3\ m$
m	Disk mass	$0.01\ kg$
g	Gravity constant	$9.81\ m/s^2$
E	Young's modulus	$2.1 \cdot 10^{11}\ Pa$
I_b	Area moment of inertia of disk	$3.97 \cdot 10^{-12}\ m^4$
$I_{\xi\xi}$	Area moment of inertia of backbone	$3.06 \cdot 10^{-7}\ m^4$

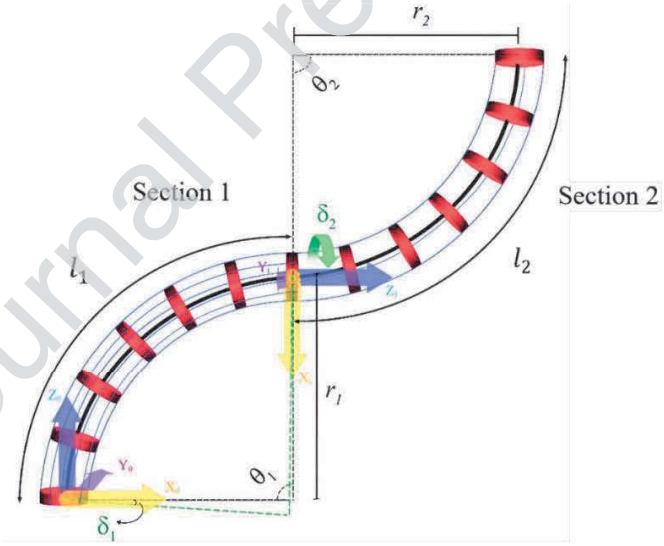

Fig. 1. Schematic of the two-segment continuum robot [19].

2.1. Kinematic Model

Under the piecewise constant curvature (PCC) assumption, the endpoint position of segment i, expressed in the preceding frame *(i-1)* is :

$$P_i^{i-1} = [x_i \quad y_i \quad z_i]^T, \quad i = 1,2 \tag{1}$$

with:

$$x_i = \frac{l_i}{\theta_i}(1-\cos(\theta_i))\cos(\delta_i), \qquad y_i = \frac{l_i}{\theta_i}(1-\cos(\theta_i))\sin(\delta_i), \qquad z_i = \frac{l_i}{\theta_i} sin\,(\theta_i)$$

The rotation matrix from frame *(i-1)* to frame *i* is:

$$R_i^{i-1} = Rot_z(\delta_i)Rot_y(\theta_i)Rot_z(-\delta_i) = \begin{bmatrix} c_{\delta_i}^2 c_{\theta_i} + s_{\delta_i}^2 & c_{\delta_i}s_{\delta_i}(c_{\theta_i}-1) & c_{\delta_i}s_{\theta_i} \\ c_{\delta_i}s_{\delta_i}(c_{\theta_i}-1) & s_{\delta_i}^2 c_{\theta_i} + c_{\delta_i}^2 & s_{\delta_i}s_{\theta_i} \\ -c_{\delta_i}s_{\theta_i} & -s_{\delta_i}s_{\theta_i} & c_{\theta_i} \end{bmatrix} \quad (2)$$

where $c_\alpha = cos\ \alpha$ and $s_\alpha = sin\ \alpha$, with α denoting a generic angle. The global endpoint position follows by recursive composition; for two segments:

$$P_2^0 = P_1^0 + R_1^0.P_2^1 \quad (3)$$

Closed-form expressions for *(x, y, z)* and the elements of R_2^0, together with their time derivatives (linear velocities) are available in [19].

2.2. Dynamic Model

Developing the dynamic model is a critical step in controller design. The dynamic model is derived using the Euler–Lagrange formulation under the PCC assumption with zero rotation of the bending plane ($\delta_1 = \delta_2 = 0$). The Euler–Lagrange equation for the i_{th} generalized coordinate θ_i is:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_i}\right) - \frac{\partial T}{\partial \theta_i} + \frac{\partial U}{\partial \theta_i} = \mathfrak{S}_i, \quad i=1,2 \quad (4)$$

where T and U denote the total kinetic and potential energies, respectively, and $\mathfrak{S}_i$ represents generalized forces. Applying Eq. (4) yields the coupled dynamic model [19]:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + K(\theta) = Q \quad (5)$$

Here, $M \in \mathbb{R}^{2\times2}$, $C \in \mathbb{R}^{2\times2}$, and $K \in \mathbb{R}^2$ denote the inertia matrix, Coriolis matrix, and gravity vector, respectively, while $Q \in \mathbb{R}^2$ is the vector of generalized input forces. The robot dynamics can be expressed in a compact form as:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + K(\theta) = F \quad (6)$$

where $\theta, F \in \mathbb{R}^p$ with $p = 2$. By defining the state variables as $\xi_1 = \theta, \xi_2 = \dot{\theta},$ and the state vector as $\xi = [\xi_1^\top, \xi_2^\top]^\top \in \mathbb{R}^n$ (where $n = 2p$) alongside the control input $u = F$, the nonlinear function are defined as $g(\xi) = M^{-1}(\theta)$, $\varpi(\xi) = -(C(\theta,\dot{\theta})\dot{\theta} + K(\theta))$.

The system can then be expressed as:

$$\dot{\xi}_1 = \xi_2, \qquad \dot{\xi}_2 = g(\xi)(u + \varpi(\xi)) \quad (7)$$

with the measured output $\xi_o = \xi_1$. The following matrices can be defined as $\Lambda = \begin{bmatrix} O_{p\times p} & I_p \\ O_{p\times p} & O_{p\times p} \end{bmatrix}$, $G = \begin{bmatrix} O_{p\times p} \\ I_p \end{bmatrix} g(\xi)$, $S = [I_p \quad O_{p\times(n-p)}]$.

Since both u and $\varpi(\xi)$ enter the system through $M^{-1}(\theta)$, we set $G_\varpi \equiv G$. The nominal state–space form becomes:

$$\dot{\xi} = \Lambda\,\xi + Gu + G_\varpi \varpi(\xi), \quad \xi_0 = S\,\xi \quad (8)$$

where, $\xi_o \in \mathbb{R}^p$ and $\xi \in \mathbb{R}^n$ with $n = 2p$. Considering fault in the actuator and sensor channel, as well as time-delay, saturation, and external disturbances, the system in Eq. (8) is rewritten as:

$$\dot{\xi} = \Lambda\,\xi + G u_\tau + G_\varpi \varpi(\xi) + G_\omega \omega(t) + G_{fa} f_a(t),$$
$$\xi_o = S\,\xi + \Theta^T \varphi(t), \tag{9}$$

where $u_\tau = sat(u(t-\tau(t)))$ is the saturated control input with actuator delay $\tau(t)$, and $\zeta = \xi(t-h(t)) - \xi(t)$ represents the effect of sensor delay. The term $\Theta^T\varphi(t) = S\,\zeta + S_{fs} f_s(t)$ aggregates both sensor delay $h(t)$ and sensor fault $f_s(t)$. Furthermore, $\omega(t)$ and $f_a(t)$ denote external disturbances and actuator faults, respectively. Here, $G_\varpi \equiv G$ because the nonlinear term $\varpi(\xi)$ enters through the same channel as the control input via $M^{-1}(\theta)$, whereas G_ω denotes the known disturbance distribution matrix associated with $\omega(t)$. Therefore, G_ϖ and G_ω are not necessarily identical, which preserves the generality of the formulation.

Note that both actuator and sensor faults are modeled as additive faults. For simulation purposes, the input history over the initial delay interval is initialized to zero so that the delayed input $u_\tau = sat(u(t-\tau(t)))$ is well defined at the start of the simulation, while the delayed output is set to the measured output ξ_o. After this initial interval, the delayed input term remains fully active in the system dynamics and control design. Note that $\Lambda \in \mathbb{R}^{n\times n}, G \in \mathbb{R}^{n\times m}, G_\omega \in \mathbb{R}^{n\times m_1}, G_{fa} \in \mathbb{R}^{n\times m_2}, S_{fs} \in \mathbb{R}^{p\times q}, S \in \mathbb{R}^{p\times n}, \Theta \in \mathbb{R}^{q\times p}$, and $\varphi(t) \in \mathbb{R}^{q\times 1}$. For the considered system $m = p$. The pair (Λ, G) is controllable, and the pair (Λ, S) is observable.

3. Mathematical Foundations and Design Procedure

This section presents the optimal control formulation, the neural approximation idea used later to implement the optimal law, and the prescribed-time design principle.

3.1. Optimal Control Formulation

Consider the following general nonlinear dynamical system:

$$\dot{\xi}(t) = \varpi(\xi) + g(\xi)u(\xi) \tag{10}$$

where $\xi \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control input, $\varpi(\xi)$ represents the drift dynamics, and $g(\xi)$ is the input gain matrix. The objective of optimal control is to determine a feedback law $u^*(\xi)$ that minimizes the infinite-horizon performance index :

$$J(\xi_0, u) = \int_t^\infty c(\xi(\tau), u(\xi(\tau)))d\tau \tag{11}$$

where $c(\xi, u) = \xi^T P_1 \xi + u^T(\xi) P_2 u(\xi) \in \mathbb{R}$ is the instantaneous cost function. The matrices $P_1 = P_1^T \in \mathbb{R}^{n\times n}$ and $P_2 = P_2^T \in \mathbb{R}^{m\times m}$ are positive semi-definite and definite, respectively.

Let $u^*(\xi)$ denote the optimal control law for the system in Eq.(10). By substituting $u(\xi)$ with $u^*(\xi)$ in Eq. (10), the optimal performance index function is defined as :

$$J^*(\xi, u) = \int_t^\infty c(\xi(\tau), u^*(\xi(\tau)))d\tau = \min_{u\in\Psi(\Omega_\xi)} \int_t^\infty c(\xi(\tau), u(\xi(\tau)))d\tau \tag{12}$$

The Hamilton–Jacobi–Bellman (HJB) equation is obtained by taking the time derivative of Eq. (12):

$$H(\xi, u^*, J^*) = \xi^T P_1 \xi + u^{*T}(\xi) P_2 u^*(\xi) + \left(\frac{\partial J^*(\xi, u)}{\partial \xi}\right)^T \times (\varpi(\xi) + g(\xi)u^*(\xi)) = 0 \tag{13}$$

Since the optimal control solution $u^*(\xi)$ associated with $J^*(\xi,u)$ is unique, the optimal performance index function in Eq. (12) is also unique. Therefore, it must satisfy the stationary condition $\partial H/\partial u^* = 0$, yielding :

$$u^*(\xi) = -\frac{1}{2}P_2^{-1}g^T(\xi)\frac{\partial J^*(\xi,u)}{\partial \xi} \tag{14}$$

Due to the uncertain gradient term $\frac{\partial J^*(\xi,u)}{\partial \xi}$, the optimal control law in Eq. (14) cannot be implemented directly. Substituting Eq.(14) into Eq.(13) leads to the HJB equation in terms of the value function gradient [39]:

$$H(\xi,u^*,J^*) = \xi^T P_1 \xi + \left(\frac{\partial J^*(\xi,u)}{\partial \xi}\right)^T \varpi(\xi) - \frac{1}{4}\left(\frac{\partial J^*(\xi,u)}{\partial \xi}\right)^T g(\xi)P_2^{-1}g^T(\xi)\frac{\partial J^*(\xi,u)}{\partial \xi} = 0 \tag{15}$$

3.2. Neural Networks

3.2.1. Radial Basis Function Neural Network-Based Dynamics Approximator

A radial basis function neural network (RBFNN) is employed to approximate the drift dynamics $\varpi(\xi)\colon \mathbb{R}^n \rightarrow \mathbb{R}^m$ as [40]:

$$\varpi(\xi) \approx W^T\Phi(\xi) \tag{16}$$

where $W \in R^{l\times m}$ is the weight matrix (so that $W^T\Phi(\xi) \in \mathbb{R}^m$), l is the number of basis functions, and $\Phi(\xi) = [\Phi_1(\xi),\dots,\Phi_l(\xi)]^T \in R^l$ is the activation vector. Each activation $\Phi_i(\xi)$ is defined as a Gaussian radial basis function, $\Phi_i(\xi) = exp\left[-\frac{\|(\xi-m_i)\|^2}{\varrho_i^2}\right]$, $i = 1,2,\dots,l$,with center $m_i \in \mathbb{R}^n$ and width $\varrho_i > 0$. The optimal NN weight W^* is defined as $W^* \triangleq \arg\min_{W\in R^l}\left\{\sup_{\xi\in\Omega_\xi} \| \varpi(\xi) - W^T\Phi(\xi) \|\right\}$, where $\Omega_\xi \subset \mathbb{R}^n$ is a compact set. Using the optimal NN weights W^*, the function $\varpi(\xi)$ can be expressed as:

$$\varpi(\xi) = W^{*T}\Phi(\xi) + \varepsilon(\xi) \tag{17}$$

where the approximation error $\varepsilon(\xi) \in \mathbb{R}^m$ is bounded and can be made arbitrarily small by increasing l [40].

3.2.2. Deep Neural Network–Based Optimal Control Approximation

Finding the analytical solution of the HJB equation in Eq. (15) is challenging. To overcome this issue, Eq. (14) can be rewritten as follows:

$$u^*(\xi) = F(\xi) - J_0(\xi) \tag{18}$$

where $F(\xi) \in \mathbb{R}^m$ is a known designer–specified stabilizing feedback (baseline term), and $J_0(\xi) \in \mathbb{R}^m$ is an unknown compensation term and can be approximated by a deep neural network (DNN) as follows [41].

$$J_0(\xi) = \phi(\xi_a) + \delta(\xi) \tag{19}$$

where $\xi_a = [\xi^T, 1]^T \in \mathbb{R}^{n+1}$ augments the state vector with a constant bias term, $\phi(\xi_a) \in \mathbb{R}^m$ is the DNN output, and $\delta(\xi)$ is the bounded DNN approximation error, satisfying $\|\delta(\xi)\| \leq \bar{\delta}$ for all $\xi \in \Omega_\xi$. The DNN consists of $\kappa+1$ layers (including the input layer), with ideal weights $\vartheta_j \in \mathbb{R}^{l_j \times l_{j+1}}$, and activation functions $\psi_j: \mathbb{R}^{l_j} \to \mathbb{R}^{l_{j+1}}$ for $j = 0,1,\ldots,k$ with $l_0 = n+1$ and $l_k = m$. The network output $\phi(\xi_a)$ is expressed as :

$$\phi(\xi_a) = \vartheta_\kappa^T \psi_\kappa \circ \cdots \circ \vartheta_1^T \psi_1 \circ \vartheta_0^T \xi_a \tag{20}$$

where $\vartheta_\kappa \in \mathbb{R}^{l_\kappa \times n}$, $\psi_\kappa \in \mathbb{R}^{l_\kappa}$,and $\phi(\xi_a) = \phi_\kappa$ is the last layer output of the DNN. The j_{th} layer output of DNN is defined as:

$$\phi_j = \begin{cases} \vartheta_j^\top \psi_j(\phi_{j-1}) & j = 1,2,\ldots,\kappa \\ \vartheta_0^\top \xi_a, & j = 0, \end{cases} \tag{21}$$

Since the ideal weights ϑ_j are unknown, they are estimated adaptively through an update law [42]. Due to the presence of sensor delay or noise, ξ may be unavailable, hence an observer is designed to provide the estimated state $\hat{\xi}$ with prescribed convergence time. Therefore, the output of the DNN defined in Eq.(20) is adaptively estimated as follows:

$$\hat{\phi}(\hat{\xi}_a) = \hat{\vartheta}_\kappa^\top \hat{\psi}_\kappa \circ \cdots \circ \hat{\vartheta}_1^\top \hat{\psi}_1 \circ \hat{\vartheta}_0^\top \hat{\xi}_a \tag{22}$$

where $\hat{\xi}_a = \left[\hat{\xi}, 1\right]^T$ denotes the augmented input vector. Here, $\hat{\vartheta}_j$ are the adaptive estimates of the weight vectors ϑ_j, and $\hat{\psi}_j = \psi_j(\hat{\phi}_{j-1})$ represents the activation function output at the j_{th} layer. The intermediate outputs $\hat{\phi}_j$ are recursively computed as:

$$\hat{\phi}_j = \begin{cases} \hat{\vartheta}_j^\top \psi_j(\hat{\phi}_{j-1}), & j = 1,2,\ldots,\kappa \\ \hat{\vartheta}_0^\top \hat{\xi}_a, & j = 0 \end{cases} \tag{23}$$

For notational simplicity, true and estimated activations can be defined as $\psi_j = \psi_j(\phi_{(j-1)})$ and $\hat{\psi}_j = \hat{\psi}_j(\hat{\phi}_{(j-1)})$, respectively. The DNN estimation error for each layer can be expressed by:

$$\tilde{\phi}_j = \phi_j - \hat{\phi}_j = \vartheta_j^\top \psi_j - \hat{\vartheta}_j^\top \hat{\psi}_j = \vartheta_j^\top \hat{\psi}_j - \hat{\vartheta}_j^\top \hat{\psi}_j + \vartheta_j^\top \psi_j - \vartheta_j^\top \hat{\psi}_j = \tilde{\vartheta}_j^\top \hat{\psi}_j + \vartheta_j^\top (\psi_j - \hat{\psi}_j) \tag{24}$$

Since $\psi_j(\phi_{(j-1)})$ is generally unknown, it is approximated using a first-order Taylor expansion around $\hat{\phi}_{(j-1)}$:

$$\psi_j(\phi_{(j-1)}) \approx \psi_j(\hat{\phi}_{(j-1)}) + \hat{\psi}'_{ji}\tilde{\phi}_{(j-1)} + O^2(\tilde{\phi}_{(j-1)}) \approx \hat{\psi}_j + \hat{\psi}'_j\tilde{\phi}_{(j-1)} + O(\|\tilde{\phi}_{(j-1)}\|^2) \tag{25}$$

where $\hat{\psi}'_j = \frac{\partial}{\partial \hat{\phi}_{(j-1)}} \psi_j(\hat{\phi}_{(j-1)})$ and $O(\|\tilde{\phi}_{(j-1)}\|^2)$ denotes higher-order terms. Thus, the difference between the true and estimated activations is $\psi_j - \hat{\psi}_j = \hat{\psi}'_j\tilde{\phi}_{(j-1)} + O(\|\tilde{\phi}_{(j-1)}\|^2)$ Substituting into Eq. (24), the layer error becomes $\tilde{\phi}_j = \tilde{\vartheta}_j^\top \hat{\psi}_j + \vartheta_j^\top \hat{\psi}'_{ji}\tilde{\phi}_{(j-1)} + \Delta_{\phi_j}$, with higher-order remainder $\Delta_{\phi_j} = \vartheta_j^\top O(\|\tilde{\phi}_{(j-1)}\|^2)$. Hence, the error associated with the j_{th} layer can then be reformulated as:

$$\tilde{\phi}_j = \left(I_{l_{j+1}} \otimes \hat{\psi}_j^\top\right) Vec(\tilde{\vartheta}_j) + \vartheta_j^\top \hat{\psi}'_j \tilde{\phi}_{(j-1)} + \Delta_{\phi_j} \tag{26}$$

For the input layer ($j=0$), this reduces to $\tilde{\phi}_0 = \left(I_{l_1} \otimes \xi_a^T\right) Vec(\tilde{\vartheta}_0^T)$. By recursively applying Eq. (26) for $j=0,1,2,\ldots,k$, the overall DNN approximation error is written in compact form as (see [41, Lemma 1]) :

$$\phi(\xi_a) - \hat{\phi}(\hat{\xi}_a) = \tilde{\phi}_\kappa = \sum_{j=0}^{\kappa} \Psi_j Vec(\tilde{\vartheta}_j) + \sum_{j=1}^{\kappa} \Xi_{\phi_j} \Delta_{\phi_j} \tag{27}$$

where $\Psi_j \in \mathbb{R}^{n\times(l_j l_{j+1})}$ are regression matrices defined by $\Psi_j = \Xi_{\xi_j}\left(I_{l_{j+1}} \otimes \hat{\psi}_j^\top\right)$, *j=0,1,2,....,k* with the initial case $\Psi_0 = \Xi_{\xi_0}\left(I_{l_1} \otimes \xi_a^T\right)$ and $\Xi_{\xi_j} \in \mathbb{R}^{n\times l_{j+1}}$. Furthermore, the propagation matrices are given by $\Xi_{\phi_j} = \prod_{\jmath=j+1}^{k} \vartheta_\jmath^\top \hat{\psi}'_\jmath = \vartheta_\kappa^\top \hat{\psi}'_\kappa \vartheta_{(\kappa-1)}^\top \hat{\psi}'_{(\kappa-1)} \cdots \vartheta_{(j+1)}^\top \hat{\psi}'_{(j+1)}$.

Since the activation functions ψ_j, optimal weights ϑ_j, and states estimation error e_ξ are bounded therefore there is a positive constant ϵ bound on $\left\|\sum_{j=1}^{\kappa} \Xi_{\phi_j}\Delta_{\phi_j}\right\|$ i.e. $\left\|\sum_{j=1}^{\kappa} \Xi_{\phi_j}\Delta_{\phi}\right\| \leq c_1$ and in a similar manner $\|\phi(\xi_a)\| \leq \bar{\phi}$ [41, 42].

3.3. Prescribed-time stability

Definition 1 [44]: Consider the nonlinear system $\dot{\xi} = f(\xi, t, d)$, where ξ denotes the state vector and $d(t)$ is a bounded disturbance. The system is prescribed-time input-to-state stable (PT-ISS) over the interval $[t_0, t_0 + T_p)$, with $T_p > 0$,if there exist a class $\kappa\mathcal{L}$ function β and a class κ function γ such that:

$$|\xi(t)| \leqslant \beta\left(|\xi(t_0)|, \rho_1(t, t_0)\right) + \gamma\left(\|d(\tau)\|_{\tau\in[t_0,t]}\right) \tag{28}$$

With the time-base generator ρ and its integral $\rho_1$defined by

- $\rho_1(t, t_0) = \int_{t_0}^{t} \rho(\tau, t_0) d\tau, \quad \rho(t, t_0) \geqslant 1$
- $\rho_1(t, t_0) \to +\infty$ monotonically as $t \to t_0 + T_p$.

Many functions satisfy the above condition. For instance, by selecting:

$$\rho(t, t_0) = k_\rho \left(\frac{T_p}{T_p + t_0 - t}\right)^{\mathscr{g}}, \tag{29}$$

$$\rho_1(t, t_0) = k_\rho \frac{T_p}{\mathscr{g} - 1}\left[\left(\frac{T_p}{T_p + t_0 - t}\right)^{\mathscr{g}-1} - 1\right] \tag{30}$$

where $k_\rho > 0$ is a scaling constant, and $\mathscr{g} \geqslant 2$ is a design parameter that adjusts the convergence rate. Therefore, $\rho_1(t, t_0) \to \infty$ as $t \to t_0 + T_p$, thereby enforcing prescribed-time convergence.

3.4. Novel Prescribed-Time Fault Observer

This section presents an adaptive observer for reconstructing the system states from outputs affected by actuator/sensor faults, delays, saturation, and external disturbances. A distinctive feature of the proposed design is the use of scaling-based terms in the adaptation laws, which enforce prescribed-time stability. Acting as high-gain forgetting factors, these terms dynamically regulate the adaptation process, enabling fast convergence and robustness against uncertainties [45]. The observer design is developed based on the state-space model in Eq. (9):

$$E\dot{\bar{\xi}} = \bar{\Lambda}\bar{\xi} + \bar{G}u_\tau + \bar{G}_\omega\omega(t) + \bar{G}_{fa}f_a(t) + \bar{G}_{\varpi}\varpi(\xi) + \bar{N}\xi_{fs} \tag{31}$$

where $\bar{\xi} = [\xi^\top, \xi_{fs}^\top]$ is the system state, $\xi_{fs} = \Theta^T \varphi(t)$ denotes sensor faults/delay from Eq.(9), $f_a(t)$ captures actuator faults/noise, $\omega(t)$ denotes the exogenous disturbances, $E = \begin{bmatrix} I_n & 0 \\ 0 & O_{p\times p} \end{bmatrix}$ is the descriptor matrix, $\bar{\Lambda} = \begin{bmatrix} \Lambda & 0 \\ 0 & -I_p \end{bmatrix}$ is the system matrix, $\bar{G} = [G^\top, O_{m\times p}]^\top$, $\bar{G}_\omega = [G_\omega^\top, O_{m_1\times p}]^\top, \bar{G}_{fa} = [G_{fa}^\top, O_{m_2\times p}]^\top, \bar{G}_\varpi = [G_\varpi^\top, O_{m_3\times p}]^\top$ are known input/disturbance/fault distribution matrices, and , $\bar{N} = [O_{p\times n}, I_p]^\top$ is the fault distribution matrix. Since directly designing an observer for Eq. (31) is nontrivial because of its nonsingular structure, a new variable is introduced: $\pi = \bar{\xi} - H^{-1}\mathcal{T}\xi_o$, where $H = E + \mathcal{T}\bar{S}$, $\bar{S} = [S, I_p]$ where $\mathcal{T} = [0_{p\times n}, Y^\top]^\top$ is the design matrix, and $Y = diag\{Y_1, Y_2, \dots, Y_p\}$, with $Y_j > 0$ for $j = 1,2, \dots, p$. Thus, the transformed state-space model can be expressed as:

$$\begin{aligned} \dot{\pi} &= \bar{\Lambda}_L\bar{\xi} - \rho_o\Pi\bar{\xi} + H^{-1}L\xi_o + \bar{G}_1u_\tau + \bar{G}_2\omega(t) + \bar{G}_3f_a(t) + \breve{\varpi}(\bar{\xi}, u_\tau) + \bar{M}\xi_{fs} \\ &= \bar{\Lambda}_L\bar{\xi} - \rho_o\Pi\bar{\xi} + H^{-1}L\xi_o + \bar{G}_1u_\tau + \bar{G}_2\omega(t) + \bar{G}_3f_a(t) + W^\top\Phi(\bar{\xi}, u_s) \\ &\quad + \bar{\varepsilon}(\bar{\xi}, u_\tau, u_s) + \bar{M}\xi_{fs} \end{aligned} \tag{32}$$

Where, $\bar{\Lambda}_L = H^{-1}(\bar{\Lambda} - L\bar{S}), \bar{G}_u = H^{-1}\bar{G} - \bar{G}_1, \bar{G}_1 = [G_1^\top, O_{m\times p}]^\top, \bar{G}_2 = H^{-1}\bar{G}_\omega, \bar{G}_3 = H^{-1}\bar{G}_{fa}$, $\bar{M} = H^{-1}\bar{N}$, and the term $\Pi > 0$ is a prescribed-time design matrix. L is a design gain that is determined so that $\bar{\Lambda}_L$ is Hurwitz. The term $\rho_o\Pi\bar{\xi}$ is added to ensure the prescribed-time stability, where $\rho_o(t)$ is the prescribed-time scaling from Eq. (29). An RBFNN term approximates the composite $\breve{\varpi}(\bar{\xi}, u_\tau) = \bar{G}_\varpi\varpi(\xi) + \rho_o\Pi\bar{\xi} + \bar{G}_u u_\tau$ with $\breve{\varpi}(\bar{\xi}) \approx W^\top\Phi(\bar{\xi}, u_s) + \bar{\varepsilon}(\bar{\xi}, u_\tau, u_s)$, $\bar{\varepsilon}(\bar{\xi}, u_\tau, u_s) \approx \bar{\varepsilon}(\bar{\xi}, u_\tau) + W^\top\left(\Phi(\bar{\xi}, u_\tau) - \Phi(\bar{\xi}, u_s)\right)$. Based on Eq. (32), the following novel adaptive prescribed-time observer is proposed:

$$\dot{\hat{\pi}} = \bar{\Lambda}_L\hat{\bar{\xi}} - \rho_o\Pi\hat{\bar{\xi}} + H^{-1}\hat{L}\xi_o + \bar{G}_1u_s + \bar{G}_1v_1 + \bar{G}_2v_2 + \bar{G}_3v_3 + \widehat{W}^T\Phi(\hat{\bar{\xi}}, u_s) + H^{-1}v_\varepsilon + \bar{M}v_4 \tag{33}$$

$$\hat{\bar{\xi}} = \hat{\pi} + H^{-1}\mathcal{T}\xi_o \tag{34}$$

$$u_s = sat(u(t)) \tag{35}$$

where, $\hat{\bar{\xi}}, \hat{L}$, and $\hat{W}$ denote the estimates of the state $\bar{\xi}$, the gain matrix L, and NN weight matrix W, respectively. Note that, in the proposed observer, u_s is used instead of u_τ because the input delay is assumed to be unknown. The compensation terms $v_1, v_2, v_3, v_\varepsilon$, and v_4 are introduced to cope with the unknown bounds of the delay mismatch, disturbances, actuator faults, approximation errors, and sensor fault/delay terms, respectively Specifically, these terms correspond to $\tilde{u} = u_s - u_\tau, \omega(t), f_a(t)$, $\tilde{\varepsilon}$, and ξ_{fs}. The following bounds are assumed: $\|\tilde{u}\| \leqslant \rho_u$, $\|\omega(t)\| \leqslant \rho_\omega$, $\|f_a(t)\| \leqslant \rho_f$, $\left\|\tilde{\varepsilon}(\bar{\xi}, \hat{\bar{\xi}}, u_\tau, u_s)\right\| \leqslant \rho_\varepsilon$, $\|\xi_{fs}\| \leqslant \rho_{fs}$.The term $\tilde{\varepsilon}(\bar{\xi}, \hat{\bar{\xi}}, u_\tau, u_s)$ will be defined in Appendix D. Based on the estimation error $\bar{e} = \hat{\bar{\xi}} - \bar{\xi} = \hat{\pi} - \pi$, the compensation terms are proposed as:

$$v_1 = -\hat{\rho}_u sign(Q_1\bar{S}\bar{e}) \tag{36}$$

$$v_2 = -\hat{\rho}_\omega sign(Q_2\bar{S}\bar{e}) \tag{37}$$

$$v_3 = -\hat{\rho}_f sign(Q_3 \bar{S}\bar{e}) \tag{38}$$
$$v_\varepsilon = -\hat{\rho}_\varepsilon sign(Q_\varepsilon \bar{S}\bar{e}) \tag{39}$$
$$v_4 = -\hat{\rho}_{fs} sign(Q_4 \bar{S}\bar{e}) \tag{40}$$

The adaptive update laws for the unknown bounds are given by:

$$\dot{\hat{\rho}}_u = \gamma_u \left(\left(sign(Q_1 \bar{S}\bar{e}) \right)^T Q_1 \bar{S}\bar{e} - \rho_o \hat{\rho}_u \right) \tag{41}$$
$$\dot{\hat{\rho}}_\omega = \gamma_\omega \left(\left(sign(Q_2 \bar{S}\bar{e}) \right)^T Q_2 \bar{S}\bar{e} - \rho_o \hat{\rho}_\omega \right) \tag{42}$$
$$\dot{\hat{\rho}}_f = \gamma_f \left(\left(sign(Q_3 \bar{S}\bar{e}) \right)^T Q_3 \bar{S}\bar{e} - \rho_o \hat{\rho}_f \right) \tag{43}$$
$$\dot{\hat{\rho}}_\varepsilon = \gamma_\varepsilon \left(\left(sign(Q_\varepsilon \bar{S}\bar{e}) \right)^T Q_\varepsilon \bar{S}\bar{e} - \rho_o \hat{\rho}_\varepsilon \right) \tag{44}$$
$$\dot{\hat{\rho}}_{fs} = \gamma_{fs} \left(\left(sign(Q_4 \bar{S}\bar{e}) \right)^T Q_4 \bar{S}\bar{e} - \rho_o \hat{\rho}_{fs} \right) \tag{45}$$

The observer gains and NN weights adaptation are updated as:

$$\dot{\hat{L}} = \Gamma_L \left(Q_y \bar{S}\bar{e} y^\top - \rho_o \hat{L} \right) \tag{46}$$
$$\dot{\hat{W}} = \Gamma_w \left(\Phi(\hat{\bar{\xi}}, u_s) \bar{e}^\top \bar{S}^\top Q_\varpi^T - \rho_o \hat{W} \right) \tag{47}$$

$\gamma_u, \gamma_\omega, \gamma_f, \gamma_\varepsilon, \gamma_{fs} > 0$ are positive constants, Γ_L and Γ_w are positive definite matrices, and $\rho_o = k_o \left(\frac{T_{p,o}}{T_{p,o} + t_0 - t} \right)^{g_o}$, $g_o \geq 2$ ensure prescribed-time convergence.

Theorem 1: The designed observer Eq. (33), utilizing the compensation terms Eq.(36)–Eq.(40) and the adaptation laws Eq. (41)–Eq. (47), guarantees prescribed-time convergence of the estimation error and reconstructs the system states and sensor-fault components from delayed/compromised outputs within the prescribed time T_p, provided that positive definite matrices $\mathcal{P}$ and Q exist satisfying:

$$H^{-\top} \mathcal{P} = Q_\varepsilon \bar{S} = Q_y \bar{S} \tag{48}$$
$$\bar{G}_1^\top \mathcal{P} = Q_1 \bar{S} \tag{49}$$
$$\bar{G}_2^\top \mathcal{P} = Q_2 \bar{S} \tag{50}$$
$$\bar{G}_3^\top \mathcal{P} = Q_3 \bar{S} \tag{51}$$
$$\bar{M}^T \mathcal{P} = Q_4 \bar{S} \tag{52}$$
$$\mathcal{P} = Q_\varpi \bar{S} \tag{53}$$
$$\mathcal{P} \bar{\Lambda}_L + \bar{\Lambda}_L^\top \mathcal{P} = -Q < 0 \tag{54}$$

where $Q_\varpi = H Q_y$.

Proof: Refer to the Appendix D.

3.5. Novel Composite Deep Learning-Based Proximate Adaptive Optimal Control

This section develops a composite adaptive learning-based optimal controller that minimizes both the tracking error and the control effort within the optimal control framework introduced in Section 3.1. To guarantee closed-loop stability, prescribed-time convergence, and robustness against model uncertainties, the control law combines analytically derived compensators with a deep

neural network (DNN). Because the exact optimal control policy cannot be computed explicitly, the DNN approximates its residual online, utilizing the real-time adaptation mechanism detailed in Section 3.2.2.

Fig. 2 shows the overall block diagram of the proposed control architecture for the surgical continuum robot. It includes the reference trajectory generator, the prescribed-time descriptor observer (Eqs. (41)– (47)), the DNN-based optimal policy approximator (Eq. (75)), the radial basis function neural network for fault approximation, the auxiliary saturation compensator (Eq. (78)), and the adaptation mechanism (Eq. (79)) that combines tracking and estimation information to generate the final control input.

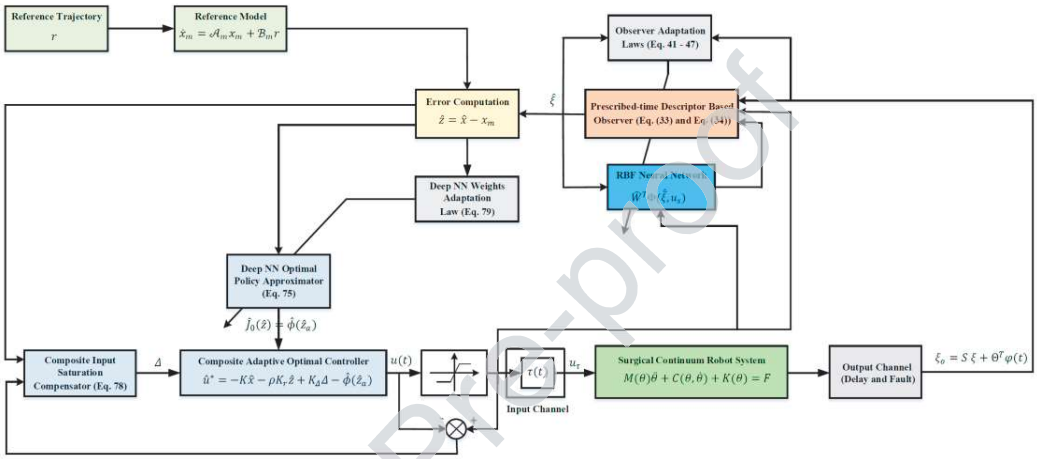

Fig. 2: Block diagram of the proposed composite adaptive optimal control

The system model Eq. (9) can be expressed as:

$$\dot{\xi} = \Lambda\xi + G_1 u + G_1\Delta u - G_1\tilde{u} + G_\omega\omega(t) + G_{fa}f_a(t) + \mathcal{W}(\xi, u_\tau) \tag{55}$$

where $\Delta u = u_s - u$ represents the input saturation mismatch, and $\tilde{u}$=u_s−$u_\tau$denotes the input − delay mismatch, *and* $\mathcal{W}(\xi, u_\tau) = G_{\varpi}\varpi(\xi) + (G - G_1)u_\tau$ represents unknown nonlinear terms. To enhance tracking, an integral error state is introduced:

$$x = [\xi^T, \sigma^T]^T, \qquad \dot{\sigma} = K_I\left(S\hat{\xi} - r\right) \tag{56}$$

where $\hat{\xi} = [I_n \quad 0_{n\times p}]\hat{\bar{\xi}}$ is the observer state and r is the reference. The extended dynamics become:

$$\dot{x} = \mathcal{A}x + \mathcal{B}u + \mathcal{B}\Delta u - \mathcal{B}\tilde{u} + \mathcal{B}_\omega\omega(t) + \mathcal{B}_{fa}f_a(t) + \mathcal{B}_w\mathcal{W}(\xi, u_\tau) + \mathcal{B}_m\, r - \breve{K}_I e_\xi \tag{57}$$

where $\mathcal{A} = \begin{bmatrix} \Lambda & O_{n\times p} \\ K_I S & O_{p\times p} \end{bmatrix}$, $\mathcal{B} = \begin{bmatrix} G_1 \\ O_{p\times m} \end{bmatrix}$, $\mathcal{B}_\omega = \begin{bmatrix} G_\omega \\ O_{p\times m_1} \end{bmatrix}$, $\mathcal{B}_{fa} = \begin{bmatrix} G_{fa} \\ O_{p\times m_2} \end{bmatrix}$, $\mathcal{B}_w = \begin{bmatrix} I_n \\ O_{p\times n} \end{bmatrix}$, $\mathcal{B}_m = \begin{bmatrix} O_{n\times p} \\ -K_I I \end{bmatrix}$, $\breve{K}_I = \begin{bmatrix} O_{n\times n} \\ K_I S \end{bmatrix}$, $e_\xi = (\xi - \hat{\xi})$.

Thus, the augmented system becomes:

$$\dot{x} = \mathcal{A}x + \mathcal{B}u + \mathcal{B}\Delta u - \mathcal{B}\tilde{u} + \mathcal{B}_\omega\omega(t) + \mathcal{B}_{fa}f_a(t) + \mathcal{B}_w\mathcal{W}(\xi, u_\tau) + \mathcal{B}_m\, r - \breve{K}_I e_\xi \tag{58}$$

Consider the model reference system as follows:

$$\dot{x}_m = \mathcal{A}_m x_m + \mathcal{B}_m r \tag{59}$$

Define the tracking error as:

$$z = x - x_m \tag{60}$$

The tracking error dynamics can be written as:

$$\dot{z} = \dot{x} - \dot{x}_m = \mathcal{A}x + \mathcal{B}u + \mathcal{B}\Delta u - \mathcal{B}\tilde{u} + \mathcal{B}_\omega \omega(t) + \mathcal{B}_{fa} f_a(t) + \mathcal{B}_\mathcal{W} \mathcal{W}(\xi, u_\tau) + \mathcal{B}_m r - \mathcal{A}_m x_m - \mathcal{B}_m r \tag{61}$$

Assume $\mathcal{A}_m = \mathcal{A} - \mathcal{B}K < 0$, so $\mathcal{A} = \mathcal{A}_m + \mathcal{B}K$, yielding:

$$\dot{z} = \mathcal{A}_m z + \mathcal{B}Kx + \mathcal{B}u + \mathcal{B}\Delta u - \mathcal{B}\tilde{u} + \mathcal{B}_\omega \omega(t) + \mathcal{B}_{fa} f_a(t) + \mathcal{B}_\varpi \mathcal{W}(\xi, u_\tau) \tag{62}$$

To ensure the prescribed-time convergence, the term $\rho\mathcal{A}_r z$ is added to Eq.(61), where ρ is the scaling factor defined in Eq.(29) and $\mathcal{A}_r = \mathcal{A} - \mathcal{B}K_r$.

$$\dot{z} = \mathcal{A}_m z + \rho\mathcal{A}_r z + \mathcal{B}Kx - \rho\mathcal{A}z + \rho\mathcal{B}K_r z + \mathcal{B}u + \mathcal{B}\Delta u - \mathcal{B}\tilde{u} + \mathcal{B}_\omega \omega(t) + \mathcal{B}_{fa} f_a(t) + \mathcal{B}_\mathcal{W} \mathcal{W}(\xi, u_\tau) \tag{63}$$

Define estimated state $\hat{x} = \left[\hat{\xi}^T, \sigma^T\right]^T, and\ \hat{z} = \hat{x} - x_m$, so $z - \hat{z} = x - \hat{x} = \mathcal{K}e_\xi$,where $e_\xi = \xi - \hat{\xi}$, denotes the state estimation error , and $\mathcal{K} = \begin{bmatrix} I_n \\ O_{p\times n} \end{bmatrix}$. Since the full state x is not available, $\hat{x}$ is used instead so that:

$$\begin{aligned} \dot{z} = {} & \mathcal{A}_m z + \rho\mathcal{A}_r z - \rho\mathcal{A}z + \mathcal{B}K\hat{x} + \rho\mathcal{B}K_r\hat{z} + \mathcal{B}u + \mathcal{B}\Delta u - \mathcal{B}\tilde{u} + \mathcal{B}_\omega \omega(t) + \mathcal{B}_{fa} f_a(t) \\ & + \mathcal{B}_\mathcal{W} \mathcal{W}(\xi, u_\tau) + \mathcal{B}K(x - \hat{x}) + \rho\mathcal{B}K_r(z - \hat{z}) \end{aligned} \tag{64}$$

Thus:

$$\begin{aligned} \dot{z} = {} & \mathcal{A}_m z + \rho\mathcal{A}_r z - \rho\mathcal{A}z + \mathcal{B}K\hat{x} + \rho\mathcal{B}K_r\hat{z} + \mathcal{B}u + \mathcal{B}\Delta u - \mathcal{B}\tilde{u} + \mathcal{B}_\omega \omega(t) + \mathcal{B}_{fa} f_a(t) \\ & + \mathcal{B}_\mathcal{W} \mathcal{W}(\xi, u_\tau) + \breve{\mathcal{K}} e_\xi \end{aligned} \tag{65}$$

where $\breve{\mathcal{K}} = (\mathcal{B}K + \rho\mathcal{B}K_r)\mathcal{K}$. Defined the quadratic performance index as:

$$J(z) = \int_t^{+\infty} (z^\top(\tau) P z(\tau) + u^T(\tau) R u(\tau)) d\tau \tag{66}$$

where P and R are positive definite matrices. The optimal control law is derived by minimizing Eq. (66):

$$J^*(z) = \min_{u \in \psi(\Omega_u)} \left(\int_t^{+\infty} (z^\top(\tau) P z(\tau) + u^\top(\tau) R u(\tau)) d\tau \right) = \int_t^{+\infty} (z^\top(\tau) P z(\tau) + u^{*T}(\tau) R u^*(\tau)) d\tau \tag{67}$$

Following the procedure in Section 3.1, the HJB equation becomes:

$$\begin{aligned} H(z, u^*, J^*) = {} & z^\top P z + u^{*\top} R u^* + \left(\frac{\partial J^*(z)}{\partial z}\right)^T \dot{z}^\top \\ = {} & z^\top P z + u^{*\top} R u^* \\ & + \left(\frac{\partial J^*(z)}{\partial z}\right)^T \left(\mathcal{A}_m z + \rho\mathcal{A}_r z - \rho\mathcal{A}z + \mathcal{B}K\hat{x} + \rho\mathcal{B}K_r\hat{z} + \mathcal{B}u + \mathcal{B}\Delta u \right. \\ & \left. - \mathcal{B}\tilde{u} + \mathcal{B}_\omega \omega(t) + \mathcal{B}_{fa} f_a(t) + \mathcal{B}_\mathcal{W} \mathcal{W}(\xi, u_\tau) + \breve{\mathcal{K}} e_\xi \right) \end{aligned} \tag{68}$$

The optimal control law is obtained by solving $\frac{\partial H(z, u^*, J^*)}{\partial u^*} = 0$, yielding:

$$u^* = -\frac{1}{2} R^{-1} \mathcal{B}^\top \frac{\partial J^*(z)}{\partial z} \tag{69}$$

Since $\frac{\partial J^*(z)}{\partial z}$ is unknown, Eq. (69) is adjusted to mitigate uncertainty, ensuring stability, prescribed-time convergence, and addressing input saturation:

$$u^* = -K\hat{x} - \rho K_r \hat{z} + K_\Delta \Delta - J_0(z) \tag{70}$$

where $\Delta \in \mathbb{R}^m$ represents the dynamic state vector of the auxiliary compensator designed to mitigate the adverse effects of input saturation and input-delay mismatch, and $K_\Delta \in \mathbb{R}^{m\times m}$ is its associated design gain matrix. Hence, the term $K_\Delta \Delta \in \mathbb{R}^m$ provides an additional dynamic corrective action that helps preserve closed-loop stability in the presence of saturation and delay effects. For the considered system, m=p=2.

$$J_0(z) = -K\hat{x} - \rho K_r \hat{z} + K_\Delta \Delta + \frac{1}{2} R^{-1} \mathcal{B}^\top \frac{\partial J^*(z)}{\partial z} \tag{71}$$

As $J_0(z)$ is unknown, a DNN approximator is used, as introduced in Section 3.2.2.

$$J_0(z) = \phi(z_a) + \delta(z) \tag{72}$$

Where $\phi(z_a) = \vartheta_\kappa^T \psi_\kappa \circ \cdots \circ \vartheta_1^T \psi_1 \circ \vartheta_0^T z_a$ with $z_a = [z^T, 1]^T$, and $\delta(z)$ is the residual approximate error; the properties of the DNN are detailed in Section 3.2.2. Based on Eqs. (69), (70), and Eq. (72), we have:

$$\mathcal{J}(z) = \frac{\partial J^*(z)}{\partial z} = 2\mathcal{B}(\mathcal{B}^\top \mathcal{B})^{-1} R(K\hat{x} + \rho K_r \hat{z} - K_\Delta \Delta + \phi(z_a) + \delta(z)) \tag{73}$$

$$u^* = -K\hat{x} - \rho K_r \hat{z} + K_\Delta \Delta - \phi(z_a) - \delta(z) \tag{74}$$

Since $\vartheta_i,\ i = 1,2,\dots,k$ is unavailable, we approximate it using a DNN:

$$\hat{\phi}(\hat{z}_a) = \hat{\vartheta}_\kappa^\top \hat{\psi}_\kappa \circ \cdots \circ \hat{\vartheta}_1^\top \hat{\psi}_1 \circ \hat{\vartheta}_0^\top \hat{\psi}_0 \tag{75}$$

With $\hat{z}_a = [\hat{z}^T, 1]^T$. Then,

$$\hat{\mathcal{J}}(\hat{z}) = 2\mathcal{B}(\mathcal{B}^\top \mathcal{B})^{-1} R\left(K\hat{z} + \rho K_r \hat{z} - K_\Delta \Delta + \hat{\phi}(\hat{z}_a)\right) \tag{76}$$

The optimal control law Eq.(74) is unknown, so its estimate is used:

$$\hat{u}^* = -K\hat{x} - \rho K_r \hat{z} + K_\Delta \Delta - \hat{\phi}(\hat{z}_a) \tag{77}$$

An adaptation law is proposed to mitigate input saturation effects while ensuring prescribed-time convergence:

$$\dot{\Delta} = \Gamma_\Delta \left(-\sigma_\Delta \Delta - \rho \Delta - \Delta \frac{\|\hat{z} P \mathcal{B} \Delta u\| + \left\| e_y^T \mathcal{M}^T \Delta u \right\| + \alpha_6 \|\mathcal{M}^T \Delta u\|^2}{\|\Delta\|^2} \right) \tag{78}$$

With $\Delta u = u_s - u$, $e_y = \xi_o - \hat{\xi}_{fs} - S\hat{\xi}$ and $\alpha_6 > 0$. The DNN weight adaptation law is designed to ensure prescribed-time convergence:

$$Vec\left(\dot{\hat{\vartheta}}_j\right) = \Gamma_{\vartheta j}\left(-\sigma_{\vartheta j} Vec\left(\hat{\vartheta}_j\right) - \rho Vec\left(\hat{\vartheta}_j\right) + \Psi_j^T \mathcal{B}^T P \hat{z} + \Psi_j^T \mathcal{M} e_y\right) \tag{79}$$

Theorem 2: The proposed composite deep learning-based proximate adaptive optimal control Eq. (77), combined with the input saturation compensator Eq. (78) and the DNN weight adaptation law Eq. (79), stabilizes the system described by Eq. (9), associated with the continuum robot defined by Eq. (6). This stabilization occurs within the prescribed time T_p to a specified radius around the origin if there exist positive definite matrices P, Q_1, and Q_2 satisfying the following equations and inequalities:

$$\mathcal{K}^T P \mathcal{B} = S^\top \mathcal{M}^T \tag{80}$$

$$Q_1 = \mathcal{A}_m^T P + P\mathcal{A}_m + \alpha_s < 0 \quad (81)$$
$$Q_2 = \mathcal{A}_r^T P + P\mathcal{A}_r - 2P\mathcal{A} + \alpha_t < 0 \quad (82)$$
$$\sigma_\Delta - \beta_3 > 0 \quad (83)$$
$$\sigma_{\vartheta j} - (\alpha_5 + \alpha_{14}) I_{l_j l_{j+1}} > 0 \quad (84)$$
$$\mathfrak{g} = 1 - \alpha_{15} > 0 \quad (85)$$

α_s and α_t are design margins and positive constants, where larger values are desirable. $\beta_3, \alpha_5, \alpha_{14}, \alpha_{15}$ are known positive constants arising from bounding cross-terms in the Lyapunov derivative (Young/Schur bounds), determined by chosen gains, P, and the known uncertainty bounds .

Proof: Refer to Appendix E.

Remark 1: Theorem 1 does not explicitly include adjustable positive scalar parameters; the observer design is fully characterized by the matrix decision variables obtained from the associated LMIs. In practical YALMIP implementations, auxiliary numerical settings (such as variable bounds, solver tolerances, and scaling factors) are introduced to ensure feasibility and numerical robustness. These empirical settings do not affect the theoretical guarantees of the theorem. In contrast, Theorem 2 incorporates several explicitly tunable design parameters (α_s, α_t, β_3, α_5, α_{14}, and α_{15}) that govern the adaptation laws of the input saturation compensator and the DNN weight update. The parameters α_s and α_t are chosen to balance LMI feasibility with robustness against system uncertainties, delays, and faults; excessively large values may render the LMIs infeasible. The remaining parameters arise from bounding arguments in the Lyapunov analysis, as detailed in Appendix E. In practice, all parameters are tuned via systematic trial-and-error to achieve a suitable trade-off among convergence speed, robustness, and closed-loop stability.

Remark 2: The stability guarantees derived in this work rely on boundedness assumptions for uncertainties and approximation errors that are explicitly incorporated into the Lyapunov analysis, rather than on exact parameter values. Parameter variations and tuning inaccuracies appear as norm-bounded perturbations in the closed-loop dynamics and are compensated by conservative LMI conditions and adaptive update laws. Consequently, provided that these variations remain within the bounds ensuring LMI feasibility, closed-loop stability is preserved, while parameter deviations primarily affect transient behavior rather than causing instability. While our proposed framework ensures convergence to a small ultimate bound, explicitly prescribing the exact size of this neighborhood via Prescribed Performance Control (PPC) [46–48] remains a highly valuable direction for future research to ensure strict safety constraints in surgical operations.

Remark 3: The proposed composite approximation, a prescribed-time baseline controller augmented by a DNN that learns only the residual costate-dependent component of the HJB solution, is designed to reconcile three conflicting requirements: analytical tractability, robustness against simultaneous faults, delays, and saturation, and real-time feasibility for surgical continuum robots. Classical MPC and direct numerical HJB solvers typically require repeated high-dimensional optimization over long horizons and are computationally prohibitive for the multi-DoF, delay-affected system considered here. Similarly, shallow neural networks, fuzzy approximators, or purely robust/sliding-mode controllers must approximate the entire optimal law and rarely achieve prescribed-time convergence under the combined adverse conditions. In

contrast, the proposed structure provides explicit LMI-based conditions for prescribed-time stability, maintains moderate computational complexity via simple forward propagation and analytic weight updates, and demonstrates substantial performance gains over a robust baseline controller in the presented simulations. It should be noted that the present optimal-control framework is developed under an additive actuator fault model, which preserves the nominal input gain matrix and ensures that the corresponding HJB equation remains well posed. Extending the proposed composite deep-learning-based optimal structure to unknown multiplicative actuator faults or loss-of-effectiveness faults, where the input matrix becomes uncertain, remains an important direction for future research.

4. Simulation and Results

This section evaluates the performance of the proposed prescribed-time fault-tolerant controller under demanding operating conditions, including actuator (input) faults, sensor (output) faults, nonlinear disturbances, parametric uncertainties, input delay, and actuator saturation. Since the target application is surgical continuum robotics, fast transient response and small steady-state error are essential to prevent performance degradation that could compromise tissue safety. The proposed approach is implemented on the two-segment continuum robot model, while the above adverse conditions are introduced at different time instants. For comparison, the robust strategy in [38] is used as a benchmark. Two reference trajectories, namely a multi-step profile and a sinusoidal profile, are considered to evaluate both transient and steady-state performance. All simulations were carried out in the Simulink environment using MATLAB R2020a on a Windows 10 platform equipped with an Intel Core i7-7600HQ processor and 8 GB DDR4 RAM.

4.1. Required Parameters

The simulations employ the two-segment continuum-robot model of [38], with physical parameters taken from Table 1 and [49]. The reference model in Eq. (59) employs the following matrices:

$$\mathcal{A}_m = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -18.27 & -3.44e-16 & -11.68 & -7.03e-16 & -10 & -4.19 \\ -3.69e-16 & -18.27 & -1.58 & -11.68 & -3.29e-16 & -10 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathcal{B}_m = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (86)$$

The proposed framework consists of three main components: a deep neural network (DNN), an observer, and a controller. The DNN contains three hidden layers with 7, 8, and 9 neurons, respectively. The activation functions are hyperbolic tangent, hyperbolic tangent, and logistic sigmoid, followed by a linear output layer. The observer and controller gains, together with the remaining design constants, are summarized in Table 2. These parameters were tuned through trial and error to ensure fast error convergence, limited transient overshoot and undershoot, and compliance with the robot's physical constraints, such as actuator saturation and mechanical limits. No optimization tool was used, since satisfactory performance was obtained after a few trials. The

numerical parameters associated with the observer and controller in Theorems 1 and 2 are reported in Appendix F.

4.2. Step Reference Trajectory

The reference signal used in this simulation is a multi-step input defined as follows: for the time interval $t = [0, 10]$, the signal value is $\pi/8$; for $t = [10, 20]$, it is $\pi/2$; and for $t = [20, 30]$, it is $\pi/4$.

Table 2. Parameters of the Observer and Controller

Observer's Parameters	Π	T_p	g_o	ρ_o	γ_ε	γ_{fs}	Γ_L	l	k_o	Γ_w	γ_u	γ_ω	γ_f
	1e-3	1	3	1	1e-3	1e-3	1e-3	10	1e-10	1e-3	1e-3	1e-6	1e-3
Controller's Parameters	g	k	K_I	$\sigma_{\vartheta j}$	ρ	K_Δ	Γ_Δ	α_6	σ_Δ	$\Gamma_{\vartheta j}$			
	3	1	1	1	1	1	1	1	1	1			

As shown in Fig. 3, the proposed method converges significantly faster than the method in [38], with an approximately 17-fold reduction in settling time. This rapid convergence is particularly desirable for surgical robotic systems, where fast stabilization after abrupt changes is essential.

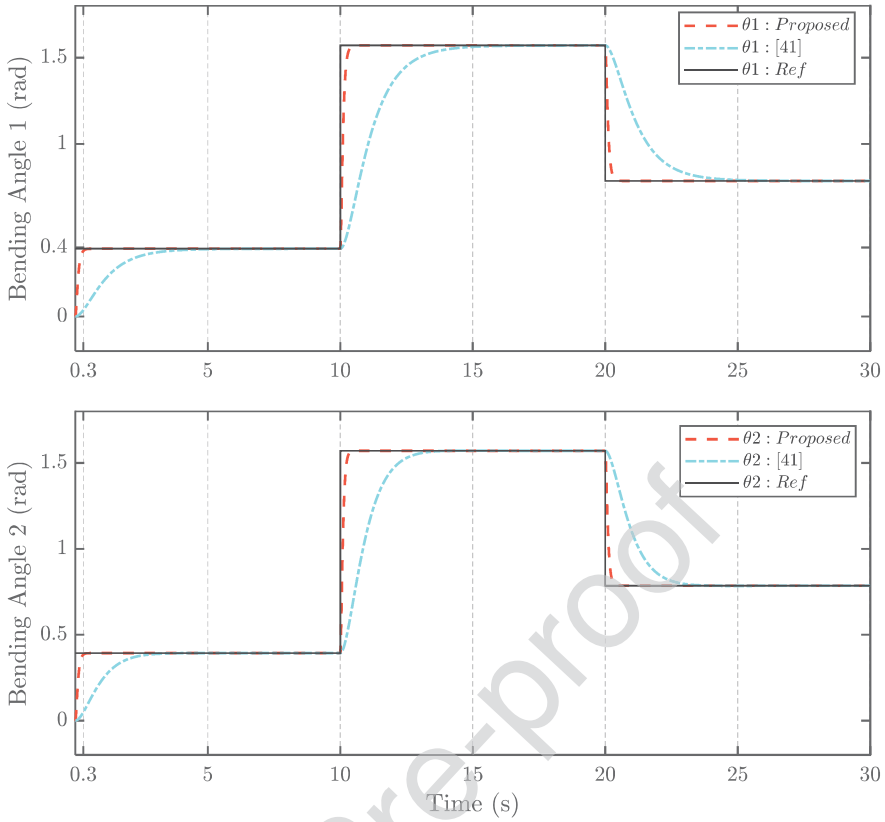

Fig. 3: Comparison of continuum robot bending angles tracking under step signal.

Fig. 4 presents the tracking error of the bending angles for the two-segment continuum robot. The proposed prescribed-time fault-tolerant controller clearly outperforms the method in [38] in both convergence speed and safety-critical transient behavior. While the controller in [38] exhibits a noticeable tracking error for nearly 5 s after each step change, the proposed controller reduces the tracking error to essentially zero within approximately 0.3 s after the reference changes at t=10 s and t=20 s. This rapid suppression is achieved despite simultaneous disturbances, model uncertainties, input saturation, input/output delays, actuator faults, and sensor faults, thereby ensuring closed-loop stability and high-precision tracking throughout the task.

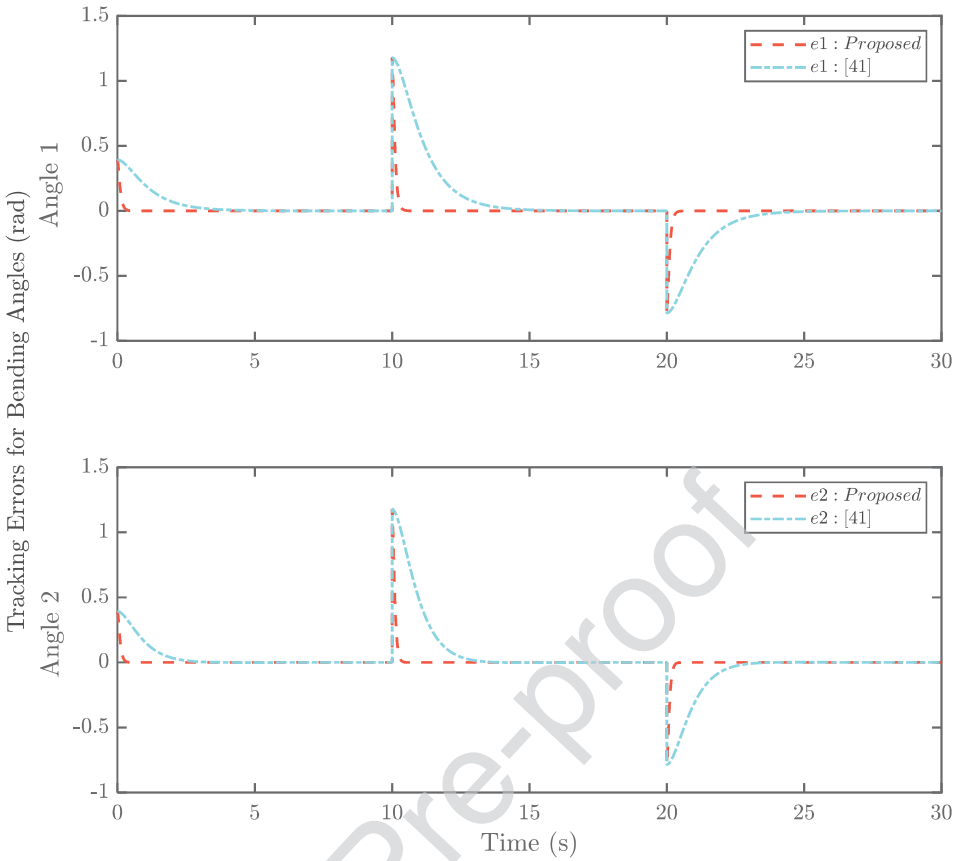

Fig. 4: Comparison of continuum robot bending angles tracking error under step signal.

Fig. 5 highlights two important observations regarding control effort. First, the proposed method achieves the desired response with substantially lower control effort than [38], which implies reduced energy consumption and less actuator wear. Second, after disturbances and faults are introduced, the control effort in [38] increases by nearly 4.4 times, whereas the proposed method maintains the control input at a moderate level, indicating effective compensation of delay, saturation, and fault effects. Fig. 6 compares the system states and their estimates. The observer accurately reconstructs the states in the presence of output-sensor faults, allowing the controller to operate with reliable feedback. The close agreement between the estimated and true states indicates robust fault accommodation and rapid estimation convergence.

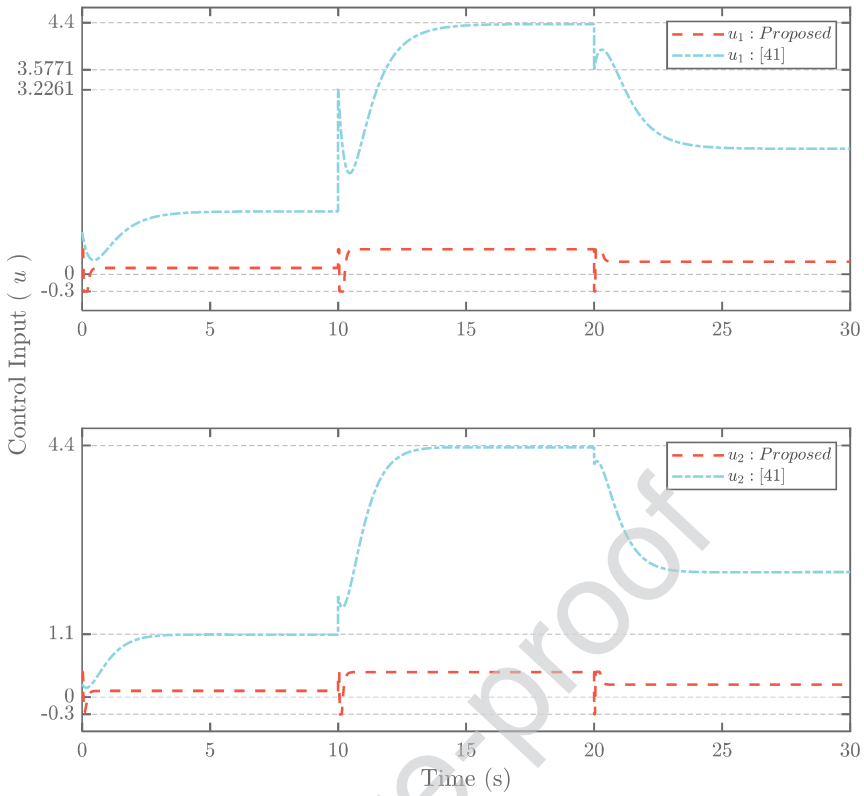

Fig. 5: Comparison of control inputs under step signal.

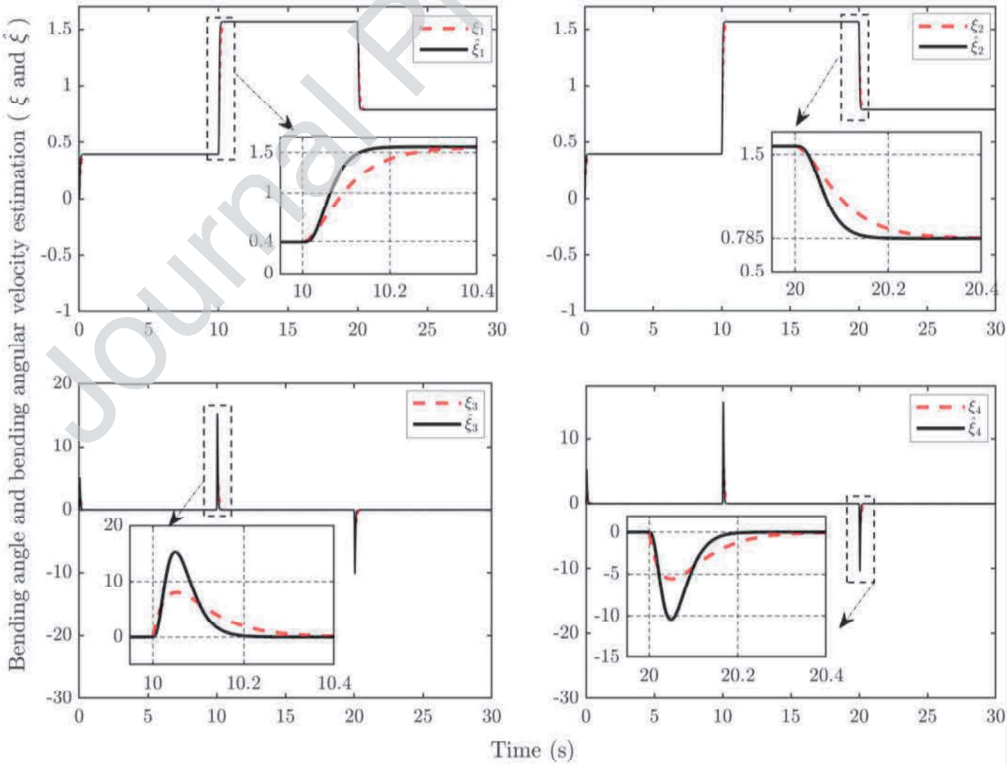

Fig. 6: Dynamic variables of the continuum robot system and their estimated values under step signal.

4.3. Sinusoidal Reference Signal

To further evaluate robustness beyond step responses, particularly in the presence of external disturbances and sensor faults, two sinusoidal reference trajectories are considered: $r_1 = \frac{\pi}{4} + \frac{\pi}{6} sin\,(t)$ and $r_2 = \frac{\pi}{4} + \frac{\pi}{6} cos\,(t)$. These waveforms also represent the tortuous vascular pathways that continuum robots are expected to follow during surgical navigation. The method in [38] is used as a representative robust benchmark, and both controllers are tested under identical conditions to ensure a fair comparison. Fig. 7 presents the tracking responses of the proposed method and the benchmark in [38] with respect to the sinusoidal reference trajectories. The proposed controller tracks the references rapidly and accurately despite the simultaneous presence of external disturbances, time delays, and injected faults. In contrast, the method in [38] exhibits clear tracking degradation under the same adverse conditions, which limits its suitability for safety-critical surgical applications. Fig. 8 shows the corresponding tracking errors. The proposed method maintains a near-zero error envelope, demonstrating rapid convergence and effective compensation of input/output faults, delays, saturation, and model uncertainties. By comparison, the benchmark in [38] produces substantially larger and more oscillatory errors under identical test conditions. Fig. 9 illustrates the control inputs. Since excessive control amplitudes may induce mechanical vibration and reduce actuator lifespan, the control signal is an important performance indicator. The proposed method generates smoother control inputs with smaller magnitude and reduced oscillation amplitude than [38], indicating lower actuator stress and improved energy efficiency. Fig. 10 compares the actual system states with their observer-based estimates. The close agreement between the estimated and true states confirms that the proposed observer can accurately reconstruct the system states even in the presence of faulty sensor measurements. This ensures that the control loop operates with reliable state information and maintains stable and accurate tracking.

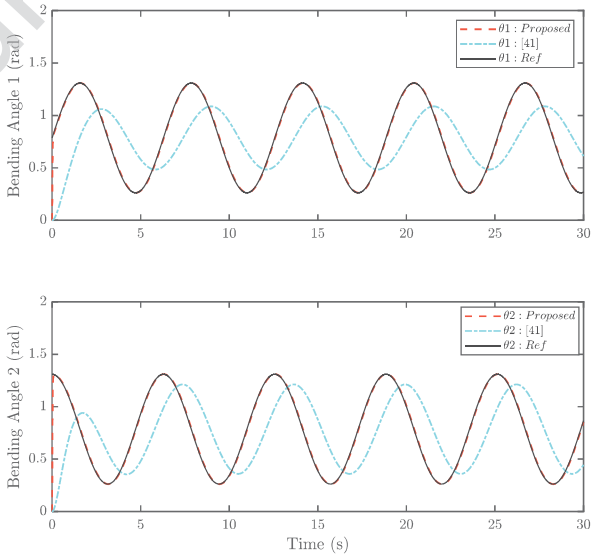

Fig. 7. Comparison of continuum robot bending angles tracking under sinusoidal reference signal.

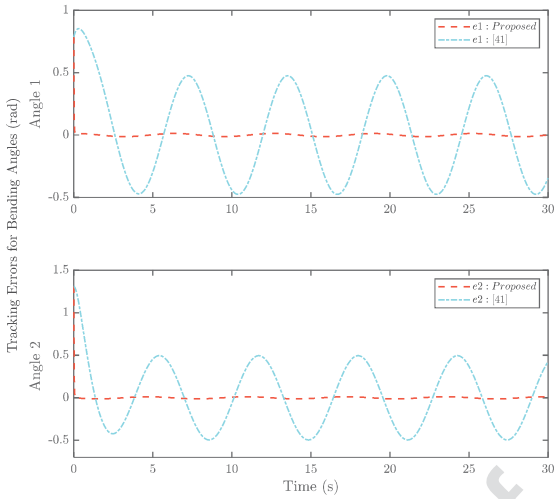

Fig. 8. Comparison of continuum robot bending angles tracking error under Sinusoidal signal.

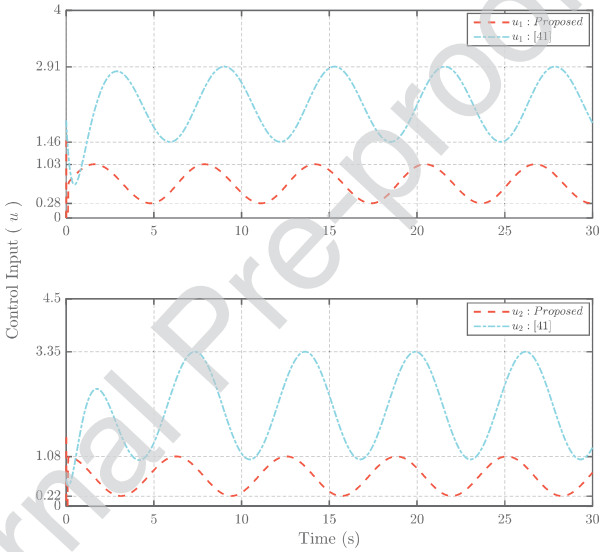

Fig. 9. Comparison of control inputs under Sinusoidal signal.

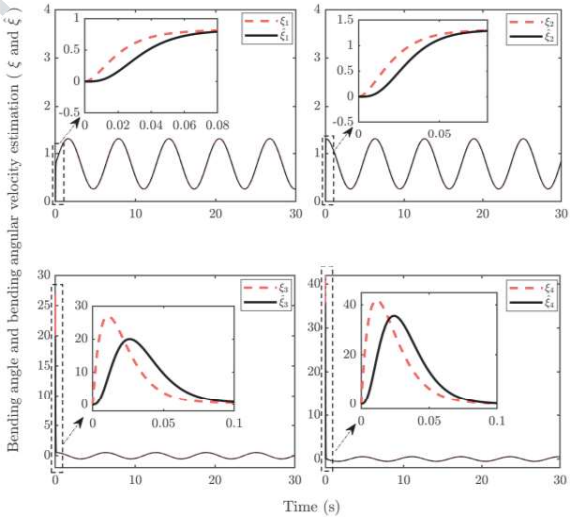

Fig. 10: System states and observer estimates under sinusoidal references.

4.4. Fault resilience

To further assess fault resilience, the proposed method is evaluated under simultaneous actuator faults, sensor faults, and time-varying input delays, and is compared with the advanced method in [50]. First, an additive actuator fault and an input delay are introduced as follows:

$$f_{ai}(t) = \begin{cases} 0, t < 20 \\ \frac{\pi}{15} + \frac{\pi}{30}\cos(3t), t \geq 20 \end{cases}, i = 1,2 \quad (87)$$

$$\tau_i(t) = 0.2 + 0.15\sin(0.5t), i = 1,2 \quad (88)$$

The actuator fault propagation matrix is defined as: $G_{fa} = [\text{zeros}(2); \text{eye}(2)]$. To evaluate the method under more challenging output-channel conditions, a sensor fault is considered as follows:

$$f_s(t) = \begin{cases} 0, t < 35 \\ \frac{\pi}{30}\sin(0.5\pi t), t \geq 35 \end{cases}, i = 1,2 \quad (89)$$

where $S_{fs} = ones(2,1)$.

Fig. 11 demonstrates the superior tracking performance of the proposed strategy under this combined fault scenario. Following the actuator fault, the proposed system recovers rapidly and returns to stable tracking within approximately 0.2 s. In contrast, the method in [50] exhibits more pronounced oscillatory behavior and requires about 0.4 s to mitigate the fault effect. This highlights the improved responsiveness and robustness of the proposed prescribed-time control scheme. In the presence of sensor faults, the proposed method also compensates effectively for degraded measurements and preserves accurate trajectory tracking. By contrast, the method in [50] suffers from noticeable tracking degradation due to the propagation of faulty feedback signals. This ability to handle both actuator and sensor faults robustly is a key advantage of the proposed approach in realistic surgical scenarios. Fig. 12 compares the system states with their corresponding observer estimates, showing that the observer successfully reconstructs the true states from corrupted measurements. Fig. 13 further illustrates the rapid detection and isolation of sensor faults, which is a central component of the overall fault-tolerant architecture.

Moreover, the integration of the DNN further enhances fault resilience by compensating for residual nonlinearities and fault-induced effects online. This predictive capability helps maintain stable operation and accurate tracking even when healthy sensor measurements become intermittently unavailable, thereby improving the overall resilience of the surgical robotic system.

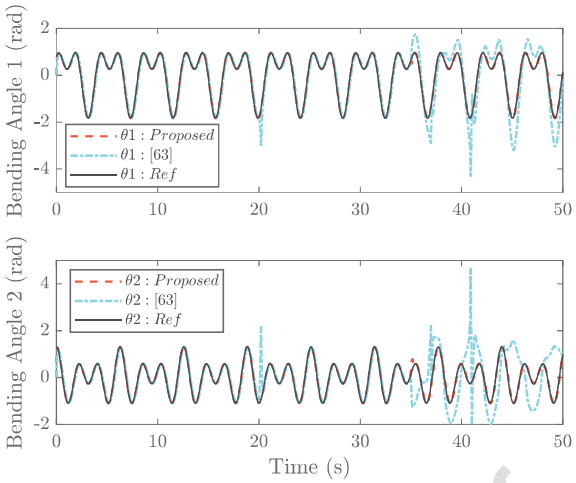

Fig. 11: Comparison of continuum robot bending angles tracking under multiple faults.

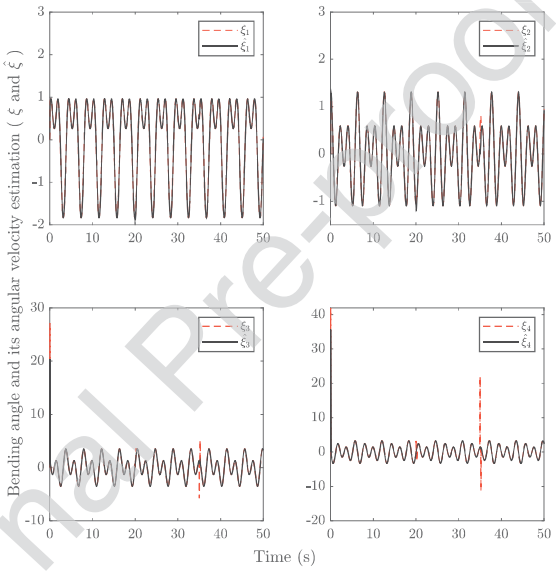

Fig. 12: System states and observer estimates under multiple faults.

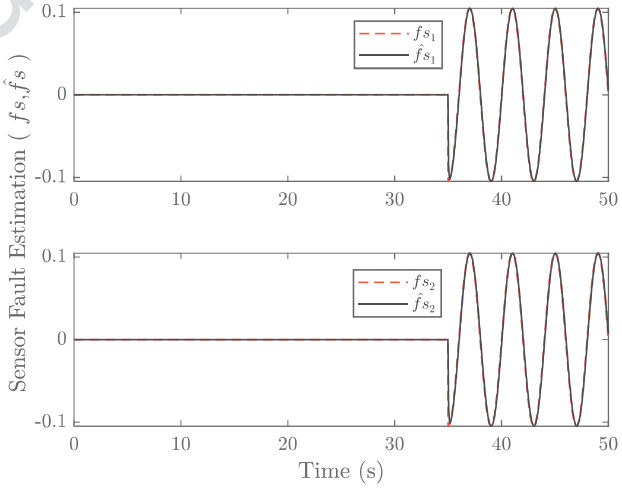

Fig. 13. Sensor faults estimation.

4.5. Parameter sensitivity analysis

The performance of the proposed controller depends on several key design parameters. To evaluate sensitivity, three scenarios were considered. In the first two scenarios, corresponding to the step and sinusoidal references, the nominal parameters listed in Table 2 were used and yielded consistently good performance. In the third scenario, involving a more complex reference trajectory together with simultaneous actuator and sensor faults and a time-varying input delay, parameter retuning was necessary to preserve prescribed-time convergence and robustness. The retuned observer parameters were selected as $T_p = 0.2\,s$, $k_o = 1.5$, $\Gamma_w = 5, l = 20, \Gamma_L = 10$, $\gamma_u = 1$, $\gamma_\omega = 1$, $\gamma_f = 1$. Reducing T_p accelerated fault estimation. Increasing k_o, l, Γ_w and Γ_L improved fault separation, state-estimation accuracy, approximation of unknown nonlinearities, and convergence of the unknown gain estimate. Increasing γ_u, γ_ω, and γ_f further enhanced robustness against disturbances, approximation errors, and fault bounds.

The retuned controller parameters are chosen as $k = 5$, $K_I = 12$, $\Gamma_{\vartheta j} = 50$ together with an expanded DNN architecture (i.e., [7, 15, 15, 15, 15, 15, 2]) compared with the lighter structure ([7, 7, 8, 9, 2]) used in the first two scenarios. Increasing k and reducing T_p accelerated control response, while increasing K_I eliminated steady-state error. In addition, increasing $\Gamma_{\vartheta j}$ and expanding the network improved optimal control approximation under faults. These adjustments were necessary to achieve fast convergence and accurate tracking under multiple adverse conditions, highlighting the trade-off among fault-estimation speed, nonlinear approximation capability, and overall robustness.

Remark 4: Although the proposed framework is developed within a rigorous theoretical setting, it remains practically feasible for real surgical continuum robots. The online implementation mainly involves structured matrix operations, observer updates, and gradient-based adaptation laws, which are well suited to modern embedded processors. Moreover, the DNNs used for approximating unmodeled dynamics and residual control terms are of moderate size and updated incrementally online, which helps keep the computational burden manageable. In practical surgical environments, measurements are inevitably affected by noise and delays. The proposed descriptor-form observer explicitly estimates delay- and fault-related effects without requiring noise-free or excessively high-frequency data, while the adaptive laws compensate for bounded disturbances and approximation errors. In addition, actuator saturation and input delays are explicitly addressed by the auxiliary compensator using the estimated states. Since the prescribed-time convergence property does not rely on restrictive initial conditions or excessively high sampling rates, the overall framework is suitable for tendon-driven surgical platforms.

5. Conclusion

This paper presented a prescribed-time fault-tolerant control framework for surgical continuum robots that simultaneously addresses actuator and sensor faults, time-varying input and output

delays, actuator saturation, external disturbances, and model uncertainties. A descriptor-form observer equipped with adaptive laws and neural-network approximation achieves prescribed-time state and fault estimation independently of the initial conditions, enabling the fast and predictable error suppression required in safety-critical surgical applications. Based on these estimates, a composite deep learning-based controller approximates online the residual term of the Hamilton-Jacobi-Bellman optimal control law, while an auxiliary compensator mitigates the effects of input saturation and delay. Rigorous Lyapunov analysis provides sufficient LMI conditions guaranteeing prescribed-time closed-loop stability. Simulations on a two-segment continuum robot demonstrated that the proposed method significantly outperforms the robust baseline controller, achieving approximately 17-fold faster settling time (0.3 s versus 5 s) and nearly threefold lower control effort, while maintaining accurate tracking and state estimation under simultaneous faults, delays, and saturation. These results highlight the strong potential of the proposed framework for safer and more precise robot-assisted surgery.

Future work will focus on experimental validation on hardware surgical platforms under realistic sensing and actuation constraints, extension to full three-dimensional continuum-robot dynamics with coupled bending and torsion effects, and incorporation of more advanced learning mechanisms to adapt the prescribed-time horizon online according to task complexity and fault severity. Additional directions include integration of perception-driven feedback from medical imaging and extension to cooperative multi-arm surgical systems with safety constraints.

References

[1] J. Burgner-Kahrs, D. C. Rucker, H. Choset, Continuum robots for medical applications: A survey, IEEE Transactions on Robotics 31 (6) (2015) 1261–1280.
[2] M. Jabari, C. Visconte, G. Quaglia, M. A. Laribi, Feasible static workspace optimization of tendon driven continuum robot based on euclidean norm, in: New Trends in Medical and Service Robots (MESROB 2025), Springer, Cham, 2025, pp. 489–501.
[3] A. Ghoul, S. Djeffal, H. Wang, K. Kara, M. L. Hadjili, Enhancing surgical robotics: A dynamic model and optimized control strategy for cable-driven continuum robots, Journal of Mechanisms and Robotics 17 (2) (2025) 021002.
[4] S. M. Ebrahimi, F. Norouzi, H. Dastres, R. Faieghi, M. Naderi, M. Malekzadeh, Sensor fault detection and compensation with performance prescription for robotic manipulators, Journal of the Franklin Institute 361 (7) (2024) 106742.
[5] T. Li, P. Huang, S. Wang, L. Qiu, C. Li, H. Ren, Modular and fault-tolerant three-axial fbg-based force sensing for transoral surgical robots, IEEE Transactions on Industrial Electronics 71 (12) (2024) 16739–16750.
[6] M. D. Fabrizio, B. R. Lee, D. Y. Chan, D. Stoianovici, T. W. Jarrett, C. Yang, L. R. Kavoussi, Effect of time delay on surgical performance during telesurgical manipulation, Journal of Endourology 14 (2) (2000) 133–138.
[7] R. K. Orosco, B. Lurie, T. Matsuzaki, E. K. Funk, V. Divi, F. C. Holsinger, S. Hong, F. Richter, N. Das, M. Yip, Compensatory motion scaling for time-delayed robotic surgery, Surgical Endoscopy 35 (2021) 2613–2618.
[8] M. Zhang, Y. Chen, Y. Gao, Y. Zhang, and Y. Zhang, “Adaptive sliding mode tracking and disturbance rejection control for cable-driven continuum robots in endoscopic surgery,” *Control Engineering Practice*, vol. 168, Art. no. 106684, 2026.
[9] S. Ling, H. Wang, P. X. Liu, Adaptive fuzzy dynamic surface control of flexible-joint robot systems with input saturation, IEEE/CAA Journal of Automatica Sinica 6 (1) (2019) 97–107.

[10] J. Duan, K. Zhang, K. Qian, J. Hao, Z. Zhang, C. Shi, Operating stiffness controller for the medical continuum robot based on impedance control, Cyborg & Bionic Systems 5 (2024) 0110.
[11] M. S. Qureshi, P. Swarnkar, S. Gupta, A supervisory online tuned fuzzy logic based sliding mode control for robotics: An application to surgical robots, Robotics and Autonomous Systems 109 (2018) 68–85.
[12] M. Ding, H. Wu, X. Zheng, Y. Guo, Adaptive predefined-time attitude stabilization control of space continuum robot, IEEE Transactions on Circuits and Systems II: Express Briefs 71 (2) (2024) 647–651.
[13] H. Liu, W. Liu, and J. Zhong, "Adaptive neural network fault-tolerant control of tendon actuated continuum robots under actuator failures and external disturbances," *Robotics and Autonomous Systems*, vol. 198, Art. no. 105310, 2026.
[14] A. Ghoul, K. Kara, M. Benrabah, B. Nasri, Control of continuum robot using two optimized pid controller, in: Proceedings of the Multi Conference on Electrical Engineering (CEE), 2021, pp. 1–6.
[15] P. Qi, C. Liu, L. Zhang, S. Wang, H.-K. Lam, K. Althoefer, Fuzzy logic control of a continuum manipulator for surgical applications, in: Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), 2014, pp. 413–418.
[16] Y. Wang, Y. Qian, W. Liu, A novel model predictive control strategy for continuum robot: Optimization and application, in: Advances in Applied Nonlinear Dynamics, Vibration, and Control – 2023, Springer Nature Singapore, 2024, pp. 484–497.
[17] Y. Song, L. Zhu, J. Li, J. Deng, C. Wang, A. Song, Mpc design of a continuum robot for pulmonary interventional surgery using koopman operators, IEEE Robotics and Automation Letters 9 (12) (2024) 10819–10826.
[18] H. Yuan, J. Zhang, Closed-loop trajectory tracking control for continuum robots: An MPC-based approach, in: Proceedings of the 2025 10th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, 2025, pp. 723–727.
[19] M. Jabari, A. Botta, L. Tagliavini, C. Visconte, G. Quaglia, A safe, high-precision reinforcement learning-based optimal control of surgical continuum robots: A monotone tube boundary approach with prescribed-time control capability, Robotics and Autonomous Systems 190 (2025) 104992.
[20] Y. S. Hagh, R. M. Asl, A. Fekih, H. Wu, H. Handroos, Active fault-tolerant control design for actuator fault mitigation in robotic manipulators, IEEE Access 9 (2021) 47912–47929.
[21] M. Ayati, P. Rezaei, H. Aghakhani, Modeling, fault detection, and stabilization of quadrotor unmanned aerial vehicle with rotor thrust deviation fault, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 237 (3) (2023) 415–432.
[22] H. Dastres, S. Mohammadreza Ebrahimi, M. Malekzadeh, F. Gordillo, Robust adaptive parameter estimator design for a multi-sinusoidal signal with fixed-time stability and guaranteed prescribed performance boundary of estimation error, Journal of the Franklin Institute 360 (1) (2023) 223–250.
[23] P. Rezaei, S. Lee, K. Cho, J. O. Hahn, Robust control of exo-abs, a wearable platform for ubiquitous respiratory assistance, Journal of Dynamic Systems, Measurement, and Control 147 (2) (2025) 021005.
[24] P. N. Paraskevopoulos, Modern Control Engineering, 1st Edition, CRC Press, Boca Raton, 2002.
[25] A. Bounemeur, M. Chemachema, and N. Essounbouli, Indirect adaptive fuzzy fault-tolerant tracking control for MIMO nonlinear systems with actuator and sensor failures, ISA Transactions 79 (2018) 45–61.
[26] A. Bounemeur and M. Chemachema, General fuzzy adaptive fault-tolerant control based on Nussbaum-type function with additive and multiplicative sensor and state-dependent actuator faults, Fuzzy Sets and Systems 468 (2023) 108616.
[27] X. Ren, Q. Guo, and T. Li, Initial condition-free prescribed performance fault-tolerant control of electro-hydraulic servo systems with state constraints, Nonlinear Dynamics 113 (10) (2025) 11723–11743.
[28] P. Wang, C. Yu, M. Lv, and Z. Zhao, "Adaptive prescribed-time neural control of nonlinear systems via dynamic surface technique," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 10, pp. 4948–4958, 2024.
[29] R. G. Brown, P. Y. C. Hwang, Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions, 3rd Edition, Wiley, New York, 1997.
[30] P. N. Dao, N. T. Dang, T. L. Nguyen, and G. K. Dinh, Finite-time sliding mode control strategies for perturbed input-constrained nonlinear bilateral teleoperation systems with variable-time communication delays, Intelligent Service Robotics (2025) 1–16.

[31] L. Li, L. Zhang, Z. Lu, and C. Yang, Neuro-Observer Based Adaptive Fixed-Time Fault-Tolerant Control for Uncertain Teleoperation System With Input Saturation and Asymmetric Time-Varying Output Constraints, International Journal of Robust and Nonlinear Control 35 (11) (2025) 4536-4553.
[32] L. Yao, M. Cui, Z. Wu, and L. Feng, "Adaptive tracking control for a class of random nonlinear systems with input delay and saturation," *Nonlinear Dynamics*, vol. 113, pp. 23167–23187, 2025.
[33] P. Wu, W. Liu, and Y. Lu, Observer-based event-triggered adaptive neural control for time-delay nonlinear systems with input saturation and external disturbances, Transactions of the Institute of Measurement and Control 45(13) (2023) 2489–2500.
[34] B. Abdelhamid and C. Mohamed, Robust fuzzy adaptive fault-tolerant control for a class of second-order nonlinear systems, International Journal of Adaptive Control and Signal Processing 39 (1) (2025) 15–30.
[35] A. Bounemeur and M. Chemachema, Adaptive fuzzy fault-tolerant control using Nussbaum-type function with state-dependent actuator failures, Neural Computing and Applications 33 (1) (2021) 191–208.
[36] O. S. Patil, D. M. Le, M. L. Greene, W. E. Dixon, Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network, IEEE Control Systems Letters 6 (2022) 1855–1860.
[37] G. Wen, C. L. P. Chen, Optimized backstepping consensus control using reinforcement learning for a class of nonlinear strict-feedback-dynamic multi-agent systems, IEEE Transactions on Neural Networks and Learning Systems 34 (3) (2023) 1524–1536.
[38] 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," *Journal of the Franklin Institute*, vol. 361, Art. no. 107129, 2024.
[39] E. Vacchini, N. Sacchi, G. P. Incremona, A. Ferrara, Design of a deep neural network-based integral sliding mode control for nonlinear systems under fully unknown dynamics, IEEE Control Systems Letters 7 (2023) 1789–1794.
[40] S. Ge, C. Hang, T. Zhang, Adaptive neural network control of nonlinear systems by state and output feedback, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29 (6) (1999) 818–828.
[41] J. Chen, J. Mei, J. Hu, Z. Yang, Deep neural networks-prescribed performance optimal control for stochastic nonlinear strict-feedback systems, Neurocomputing 610 (2024) 128633.
[42] Q. Guo, C. Wang, J. Wang, and T. Li, Reinforcement learning-based fixed-time optimal formation control for multiple mobile robots with prescribed performance, IEEE Internet of Things Journal (2025).
[43] S. Thorapalli Muralidharan, R. Zhu, Continuum actuator based soft quadruped robot, Master's thesis, KTH Royal Institute of Technology, 2020.
[44] Z. Liu, C. Lin, Y. Shang, Prescribed-time adaptive neural feedback control for a class of nonlinear systems, Neurocomputing 511 (2022) 155–162.
[45] J. Wang, C. Wang, M. Xin, Z. Ding, J. Shan, Cooperative Control of Multi-Agent Systems: An Optimal and Robust Perspective, Academic Press, 2020.
[46] Q. Xu and G. Guo, "Performance-guaranteed consensus tracking of non-smooth multiagent systems: A low-complexity design approach," *IEEE Transactions on Cybernetics*, early access, 2026.
[47] C.-L. Zhang and G. Guo, "Prescribed performance fault-tolerant control of nonlinear systems via actuator switching," *IEEE Transactions on Fuzzy Systems*, vol. 32, no. 3, pp. 1013–1022, 2024.
[48] C.-L. Zhang and G. Guo, "Approximation-free prescribed performance fault-tolerant control of nonlinear systems: Considering deferred actuator switching," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 7122–7131, 2025.
[49] A. Ghoul, K. Kara, M. Benrabah, M. L. Hadjili, Optimized nonlinear sliding mode control of a continuum robot manipulator, Journal of Control, Automation and Electrical Systems 33 (5) (2022) 1355–1363.
[50] P. Yang, S. Zhang, X. Yu and W. He, "Reinforcement-Learning-Based Finite Time Fault Tolerant Control for a Manipulator With Actuator Faults," in IEEE Transactions on Cybernetics, 55 (6), 2025, 2621-2632.
[51] Y. Song, Y. Wang, M. Krstic, Time-varying feedback for stabilization in prescribed finite time, International Journal of Robust and Nonlinear Control 29 (3) (2019) 618–633.
[52] R. F. Stengel, Optimal Control and Estimation, Courier Corporation, 1994.

Appendix C. Useful Lemmas

Lemma 1 - Young's Inequality [51]: For any given $\mathfrak{a}, \mathfrak{b} \in \mathbb{R}^n$ we have

$$2\,\mathfrak{a}^T \mathcal{S} \mathfrak{N} \mathfrak{b} \leq \mathfrak{a}^T \mathcal{S} \mathcal{p} \mathcal{S}^T a + \mathfrak{b}^T \mathfrak{N}^T \mathcal{p}^{-1} \mathfrak{N} \mathfrak{b} \tag{C.1}$$

where $\mathcal{p} > 0, \mathcal{S}$, and $\mathfrak{N}$ have appropriate dimensions.

Lemma 2 – Trace Property [52]: For real column vectors $\mathfrak{a}, \mathfrak{b} \in \mathbb{R}^n$, the trace of their outer product is equal to their inner product.

$$Tr(\mathfrak{b}\mathfrak{a}^T) \leq \mathfrak{a}^T \mathfrak{b} \tag{C.2}$$

Where Tr is the trace operator.

Lemma 3 - PT-ISS [44]: For a nonlinear system described by $\xi = f(\xi, t, d)$, if there exists a smooth, positive definite, and radially unbounded function $V: \mathbb{R}^n \to \mathbb{R}^+$ that satisfies the following conditions:

$$\dot{V}(t) \leqslant -2k^* \rho(t, t_0) V(t) + \rho(t, t_0) C_1^* + C_2^* \tag{C.3}$$

for positive constants k^*, C_1 and C_2, then

$$V(t) \leqslant V(t_0) e^{\frac{-2k^* T_p}{\vartheta - 1}\left(\left(\frac{T_p}{T_p + t_0 - t}\right)^{\vartheta - 1} - 1\right)} + \frac{C^*}{2k^*} \tag{C.4}$$

As a result, it can be established that the system achieves prescribed-time input-to-state stability (PT-ISS) within the time T_p, and the function $V(t)$ converges to $\frac{C^*}{2k^*}, t \geq T_p$. Meanwhile, $\rho(t, t_0) \geqslant 1$ and $C^* = C_1^* + C_2^*$.

Appendix D. Proof of Theorem 1

By analyzing (32) and (33), the dynamics of the observer estimation error, can be described by the following equation.

$$\begin{aligned} \dot{\bar{e}} = \dot{\bar{\xi}} - \dot{\bar{\xi}} = \dot{\hat{\pi}} - \dot{\pi} \\ = \bar{\Lambda}_L \bar{e} - \rho_o \Pi \bar{e} + H^{-1} \tilde{L} y - \bar{G}_1 \tilde{u} + \bar{G}_1 v_1 - \bar{G}_2 \omega(t) + \bar{G}_2 v_2 - \bar{G}_3 f_a(t) + \bar{G}_3 v_3 \\ - W^\top \Phi(\bar{\xi}, u_s) + \hat{W}^\top \Phi\left(\hat{\bar{\xi}}, u_s\right) - \bar{\varepsilon}(\bar{\xi}, u_\tau, u_s) + H^{-1} v - \bar{M} \xi_{fs} + \bar{M} v_4 \end{aligned} \tag{D.1}$$

Where $\tilde{u} = u_\tau - u_s$. If we define $W^\top \Phi(\bar{\xi}, u_s) + \bar{\varepsilon}(\bar{\xi}, u_\tau, u_s) = W^\top \Phi(\hat{\bar{\xi}}, u_s) + H^{-1} \tilde{\varepsilon}(\bar{\xi}, \hat{\bar{\xi}}, u_\tau, u_s)$ and $\tilde{\varepsilon}(\bar{\xi}, \hat{\bar{\xi}}, u_\tau, u_s) = H\bar{\varepsilon}(\bar{\xi}, u_\tau, u_s) + HW^\top\left(\Phi(\bar{\xi}, u_s) - \Phi(\hat{\bar{\xi}}, u_s)\right)$, we have $-W^\top \Phi(\bar{\xi}, u_s) + \hat{W}^\top \Phi\left(\hat{\bar{\xi}}, u_s\right) - \bar{\varepsilon}(\bar{\xi}, u_\tau, u_s) = (\hat{W}^\top - W^\top)\Phi\left(\hat{\bar{\xi}}, u_s\right) - H^{-1}\tilde{\varepsilon}\left(\bar{\xi}, \hat{\bar{\xi}}, u_\tau, u_s\right) = \tilde{W}^\top \Phi\left(\hat{\bar{\xi}}, u_s\right) - H^{-1}\tilde{\varepsilon}\left(\bar{\xi}, \hat{\bar{\xi}}, u_\tau, u_s\right)$.

Consider the following Lyapunov function:

$$V_1 = \bar{e}^\top \mathcal{P} \bar{e} + \frac{1}{\gamma_u}\tilde{\rho}_u^2 + \frac{1}{\gamma_\omega}\tilde{\rho}_\omega^2 + \frac{1}{\gamma_f}\tilde{\rho}_f^2 + \frac{1}{\gamma_\varepsilon}\tilde{\rho}_\varepsilon^2 + \frac{1}{\gamma_{fs}}\tilde{\rho}_{fs}^2 + Tr\left(\tilde{L}^\top \Gamma_L^{-1} \tilde{L}\right) + Tr(\tilde{W}^\top \Gamma_w^{-1} \tilde{W}) \tag{D.2}$$

Where $\tilde{\rho}_u = \hat{\rho}_u - \rho_u$, $\tilde{\rho}_\omega = \hat{\rho}_\omega - \rho_\omega$, $\tilde{\rho}_f = \hat{\rho}_f - \rho_f$, $\tilde{\rho}_\varepsilon = \hat{\rho}_\varepsilon - \rho_\varepsilon$, $\tilde{\rho}_{fs} = \hat{\rho}_{fs} - \rho_{fs}$, $\tilde{L} = \hat{L} - L$, and $\tilde{W} = \hat{W} - W$. The time derivative of V_1 is computed as follows:

$$\begin{aligned} \dot{V}_1 = \dot{\bar{e}}^\top \mathcal{P} \bar{e} + \bar{e}^\top \mathcal{P} \dot{\bar{e}} + \frac{2}{\gamma_u}\tilde{\rho}_u \dot{\hat{\rho}}_u + \frac{2}{\gamma_\omega}\tilde{\rho}_\omega \dot{\hat{\rho}}_\omega + \frac{2}{\gamma_f}\tilde{\rho}_f \dot{\hat{\rho}}_f + \frac{2}{\gamma_\varepsilon}\tilde{\rho}_\varepsilon \dot{\hat{\rho}}_\varepsilon + \frac{2}{\gamma_{fs}}\tilde{\rho}_{fs} \dot{\hat{\rho}}_{fs} + 2Tr(\tilde{L}^\top \Gamma_L^{-1} \dot{\tilde{L}}) \\ + 2Tr(\tilde{W}^\top \Gamma_w^{-1} \dot{\tilde{W}}) \end{aligned} \tag{D.3}$$

Substituting (D.1) into (D.3) yields:

$$\begin{aligned} \dot{V}_1 = \bar{e}^\top (\mathcal{P}\bar{\Lambda}_L + \mathcal{P}_L^\top P)\bar{e} - \rho_o \bar{e}^\top (\mathcal{P}\Pi + \Pi^T \mathcal{P})\bar{e} + y^\top \tilde{L}^\top H^{-\top} \mathcal{P} \bar{e} + \bar{e}^\top \mathcal{P} H^{-1} \tilde{L}^\top y - \tilde{u}^\top \bar{G}_1^\top \mathcal{P} \bar{e} + \bar{e}^\top \mathcal{P} \bar{G}_2 v_2 \\ - f_a^\top(t) \bar{G}_3^\top \mathcal{P} \bar{e} + \bar{e}^\top \mathcal{P} \bar{G}_3 f_a(t) + v_3^\top \bar{G}_3^\top \mathcal{P} \bar{e} + \bar{e}^\top \mathcal{P} \bar{G}_3 v_3 + \Phi^\top\left(\hat{\bar{\xi}}, u_s\right) \tilde{W} \mathcal{P} \bar{e} \\ + \bar{e}^\top \mathcal{P} \tilde{W}^\top \Phi\left(\hat{\bar{\xi}}, u_s\right) - \tilde{\varepsilon}^\top H^{-\top} \mathcal{P} \bar{e} - \bar{e}^\top \mathcal{P} H^{-1} \tilde{\varepsilon} + v^\top H^{-\top} \mathcal{P} \bar{e} + \bar{e}^\top \mathcal{P} H^{-1} v - \xi_{fs}^\top \bar{M}^\top \mathcal{P} \bar{e} \\ - \bar{e}^\top \mathcal{P} \bar{M} \xi_{fs} + v_4^\top \bar{M}^\top \mathcal{P} \bar{e} + \bar{e}^\top \mathcal{P} \bar{M} v_4 + \frac{2}{\gamma_u}\tilde{\rho}_u \dot{\hat{\rho}}_u + \frac{2}{\gamma_\omega}\tilde{\rho}_\omega \dot{\hat{\rho}}_\omega + \frac{2}{\gamma_f}\tilde{\rho}_f \dot{\hat{\rho}}_f + \frac{2}{\gamma_\varepsilon}\tilde{\rho}_\varepsilon \dot{\hat{\rho}}_\varepsilon \\ + \frac{2}{\gamma_{fs}}\tilde{\rho}_{fs} \dot{\hat{\rho}}_{fs} + 2Tr\left(\tilde{L}^\top \Gamma_L^{-1} \dot{\tilde{L}}\right) + 2Tr\left(\tilde{W}^\top \Gamma_w^{-1} \dot{\tilde{W}}\right) \end{aligned} \tag{D.4}$$

By applying Eqs.(47)–(53) along with Lemma 1 and Lemma 2, the following result is obtained:

$$y^\top \tilde{L}^\top H^{-\top}\mathcal{P}\bar{e} + \bar{e}^\top \mathcal{P}H^{-1}\tilde{L}^\top y = 2y^\top \tilde{L}^\top H^{-\top}\mathcal{P}\bar{e} = 2Tr\left(\tilde{L}^\top H^{-\top}\mathcal{P}\bar{e}y^\top\right) = 2Tr\left(\tilde{L}^\top \mathcal{Q}_y \bar{S}\bar{e}y^\top\right) \tag{D.5}$$

$$-\tilde{u}^\top \bar{G}_1^\top \mathcal{P}\bar{e} = -\tilde{u}^\top \mathcal{Q}_1 \bar{S}\bar{e} \le \rho_u \|\mathcal{Q}_1 \bar{S}\bar{e}\| \le \rho_u \left(sign(\mathcal{Q}_1 \bar{S}\bar{e})\right)^T \mathcal{Q}_1 \bar{S}\bar{e} \tag{D.6}$$

$$-\bar{e}^\top \mathcal{P}\bar{G}_1^\top \tilde{u} = -\bar{e}^\top \bar{S}^\top \mathcal{Q}_1^T \tilde{u} \le \rho_u (\mathcal{Q}_1 \bar{S}\bar{e})^T sign(\mathcal{Q}_1 \bar{S}\bar{e}) \tag{D.7}$$

$$-\omega^\top(t)\bar{G}_2^\top \mathcal{P}\bar{e} = -\omega^\top(t)\mathcal{Q}_2 \bar{S}\bar{e} \le \rho_\omega \|\mathcal{Q}_2 \bar{S}\bar{e}\| \le \rho_\omega \left(sign(\mathcal{Q}_2 \bar{S}\bar{e})\right)^T \mathcal{Q}_2 \bar{S}\bar{e} \tag{D.8}$$

$$-\bar{e}^\top \mathcal{P}\bar{G}_2^\top \omega(t) = -\bar{e}^\top \bar{S}^\top \mathcal{Q}_2^T \omega(t) \le \rho_\omega (\mathcal{Q}_2 \bar{S}\bar{e})^T sign(\mathcal{Q}_2 \bar{S}\bar{e}) \tag{D.9}$$

$$-f_a^\top(t)\bar{G}_3^\top P\bar{e} = -f_a^\top(t)\mathcal{Q}_3 \bar{S}\bar{e} \le \rho_f \|\mathcal{Q}_3 \bar{S}\bar{e}\| \le \rho_f \left(sign(\mathcal{Q}_3 \bar{S}\bar{e})\right)^T \mathcal{Q}_3 \bar{S}\bar{e} \tag{D.10}$$

$$-\bar{e}^\top \mathcal{P}\bar{G}_3^\top f_a(t) = -\bar{e}^\top \bar{S}^\top \mathcal{Q}_3^T f_a(t) \le \rho_f (\mathcal{Q}_3 \bar{S}\bar{e})^T sign(\mathcal{Q}_3 \bar{S}\bar{e}) \tag{D.11}$$

$$-\tilde{\varepsilon}^\top H^{-\top}\mathcal{P}\bar{e} = -\tilde{\varepsilon}^\top \mathcal{Q}_3^T \bar{S}\bar{e} \le \rho_\varepsilon \left(sign(\mathcal{Q}\bar{S}\bar{e})\right)^T \mathcal{Q}\bar{S}\bar{e} \tag{D.12}$$

$$-\bar{e}^\top \mathcal{P}H^{-1}\tilde{\varepsilon} = -\bar{e}^\top \bar{S}^\top \mathcal{Q}^T \tilde{\varepsilon} \le \rho_\varepsilon (\mathcal{Q}\bar{C}\bar{e})^T sign(\mathcal{Q}\bar{S}\bar{e}) \tag{D.13}$$

$$-\xi_{fs}^\top \bar{M}^\top P\bar{e} = -\xi_{fs}^\top \mathcal{Q}_4 \bar{S}\bar{e} \le \rho_{fs} \left(sign(\mathcal{Q}_4 \bar{S}\bar{e})\right)^T \mathcal{Q}_4 \bar{S}\bar{e} \tag{D.14}$$

$$-\bar{e}^\top \mathcal{P}\bar{M}\xi_{fs} \le \rho_{fs} (\mathcal{Q}_4 \bar{S}\bar{e})^T sign(\mathcal{Q}_4 \bar{S}\bar{e}) \tag{D.15}$$

$$\Phi^\top\left(\hat{\bar{\xi}}, u_s\right)\tilde{W}\mathcal{P}\bar{e} + \bar{e}^\top \mathcal{P}\tilde{W}^\top \Phi\left(\hat{\bar{\xi}}, u_s\right) = 2\bar{e}^\top \mathcal{P}\tilde{W}^\top \Phi\left(\hat{\bar{\xi}}, u_s\right) = 2Tr\left(\tilde{W}^\top \Phi\left(\hat{\bar{\xi}}, u_s\right)\bar{e}^\top \bar{S}^\top \mathcal{Q}_\varpi^T\right) \tag{D.16}$$

By incorporating Eqs.(36)–(39) and Eqs.(D.5)–(D.16) into the framework of Eq. (D.4), the following is derived:

$$\begin{aligned}
\dot{V}_1 \le\ & \bar{e}^\top(\mathcal{P}\bar{\Lambda}_L + \bar{\Lambda}_L^\top \mathcal{P})\bar{e} - \rho\bar{e}^\top(\mathcal{P}\Pi + \Pi^T\mathcal{P})\bar{e} + 2Tr\left(\tilde{L}^\top \mathcal{Q}_y \bar{S}\bar{e}y^\top\right) + \rho_u\left(sign(\mathcal{Q}_1\bar{S}\bar{e})\right)^T \mathcal{Q}_1\bar{S}\bar{e} \\
& + \rho_u(\mathcal{Q}_1\bar{S}\bar{e})^T sign(\mathcal{Q}_1\bar{S}\bar{e}) - \hat{\rho}_u\left(sign(\mathcal{Q}_1\bar{S}\bar{e})\right)^T \mathcal{Q}_1\bar{S}\bar{e} - \hat{\rho}_u(\mathcal{Q}_1\bar{S}\bar{e})^T sign(\mathcal{Q}_1\bar{S}\bar{e}) \\
& + \rho_\omega\left(sign(\mathcal{Q}_2\bar{S}\bar{e})\right)^T \mathcal{Q}_2\bar{S}\bar{e} + \rho_\omega(\mathcal{Q}_2\bar{S}\bar{e})^T sign(\mathcal{Q}_2\bar{S}\bar{e}) - \hat{\rho}_\omega\left(sign(\mathcal{Q}_2\bar{S}\bar{e})\right)^T \mathcal{Q}_2\bar{S}\bar{e} \\
& - \hat{\rho}_\omega(\mathcal{Q}_2\bar{S}\bar{e})^T sign(\mathcal{Q}_2\bar{S}\bar{e}) + \rho_f\left(sign(\mathcal{Q}_3\bar{S}\bar{e})\right)^T \mathcal{Q}_3\bar{S}\bar{e} + \rho_f(\mathcal{Q}_3\bar{S}\bar{e})^T sign(\mathcal{Q}_3\bar{S}\bar{e}) \\
& - \hat{\rho}_f\left(sign(\mathcal{Q}_3\bar{S}\bar{e})\right)^T \mathcal{Q}_3\bar{S}\bar{e} - \hat{\rho}_f(\mathcal{Q}_3\bar{S}\bar{e})^T sign(\mathcal{Q}_3\bar{S}\bar{e}) \\
& + 2Tr\left(\tilde{W}^\top \Phi\left(\hat{\bar{\xi}}, u_s\right)\bar{e}^\top \bar{S}^\top \mathcal{Q}_\varpi^T\right) + \rho_\varepsilon\left(sign(\mathcal{Q}\bar{S}\bar{e})\right)^T \mathcal{Q}\bar{S}\bar{e} + \rho_\varepsilon(\mathcal{Q}\bar{S}\bar{e})^T sign(\mathcal{Q}\bar{S}\bar{e}) \\
& - \hat{\rho}_\varepsilon\left(sign(\mathcal{Q}\bar{S}\bar{e})\right)^T \mathcal{Q}\bar{S}\bar{e} - \hat{\rho}_\varepsilon(\mathcal{Q}\bar{S}\bar{e})^T sign(\mathcal{Q}\bar{S}\bar{e}) + \rho_{fs}\left(sign(\mathcal{Q}_4\bar{S}\bar{e})\right)^T \mathcal{Q}_4\bar{S}\bar{e} \\
& + \rho_{fs}(\mathcal{Q}_4\bar{S}\bar{e})^T sign(\mathcal{Q}_4\bar{S}\bar{e}) - \hat{\rho}_{fs}\left(sign(\mathcal{Q}_4\bar{S}\bar{e})\right)^T \mathcal{Q}_4\bar{S}\bar{e} - \hat{\rho}_{fs}(\mathcal{Q}_4\bar{S}\bar{e})^T sign(\mathcal{Q}_4\bar{S}\bar{e}) \\
& + \frac{2}{\gamma_u}\tilde{\rho}_u\dot{\hat{\rho}}_u + \frac{2}{\gamma_\omega}\tilde{\rho}_\omega\dot{\hat{\rho}}_\omega + \frac{2}{\gamma_f}\tilde{\rho}_f\dot{\hat{\rho}}_f + \frac{2}{\gamma_\varepsilon}\tilde{\rho}_\varepsilon\dot{\hat{\rho}}_\varepsilon + \frac{2}{\gamma_{fs}}\tilde{\rho}_{fs}\dot{\hat{\rho}}_{fs} + 2Tr\left(\tilde{L}^\top \Gamma_L^{-1}\dot{\tilde{L}}\right) \\
& + 2Tr\left(\tilde{W}^\top \Gamma_w^{-1}\dot{\tilde{W}}\right)
\end{aligned} \tag{D.17}$$

$$\begin{aligned}
\dot{V}_1 \le\ & \bar{e}^\top(\mathcal{P}\bar{\Lambda}_L + \bar{\Lambda}_L^\top \mathcal{P})\bar{e} - \rho_o\bar{e}^\top(\mathcal{P}\Pi + \Pi^T\mathcal{P})\bar{e} + 2\tilde{\rho}_u\left(\frac{1}{\gamma_u}\dot{\hat{\rho}}_u - \left(sign(\mathcal{Q}_1\bar{S}\bar{e})\right)^T \mathcal{Q}_1\bar{S}\bar{e}\right) \\
& + +2\tilde{\rho}_\omega\left(\frac{1}{\gamma_\omega}\dot{\hat{\rho}}_\omega - \left(sign(\mathcal{Q}_2\bar{S}\bar{e})\right)^T \mathcal{Q}_2\bar{S}\bar{e}\right) + 2\tilde{\rho}_f\left(\frac{1}{\gamma_f}\dot{\hat{\rho}}_f - \left(sign(\mathcal{Q}_3\bar{S}\bar{e})\right)^T \mathcal{Q}_3\bar{S}\bar{e}\right) \\
& + 2\tilde{\rho}_\varepsilon\left(\frac{1}{\gamma_\varepsilon}\dot{\hat{\rho}}_\varepsilon - \left(sign(\mathcal{Q}\bar{S}\bar{e})\right)^T \mathcal{Q}\bar{S}\bar{e}\right) + \ 2\tilde{\rho}_{fs}\left(\frac{1}{\gamma_{fs}}\dot{\hat{\rho}}_{fs} - \left(sign(\mathcal{Q}_4\bar{S}\bar{e})\right)^T \mathcal{Q}_4\bar{S}\bar{e}\right) \\
& + 2Tr\left(\tilde{L}^\top\left(\Gamma_L^{-1}\dot{\tilde{L}} + \mathcal{Q}_y\bar{S}\bar{e}y^\top\right)\right) + 2Tr\left(\tilde{W}^\top\left(\Gamma_w^{-1}\dot{\tilde{W}} + \Phi\left(\hat{\bar{\xi}}, u_s\right)\bar{e}^\top \bar{S}^\top \mathcal{Q}_\varpi^T\right)\right)
\end{aligned} \tag{D.18}$$

After inserting the adaptation laws Eqs.(40)–(46) into (D.18), the following inequality is obtained:

$$\begin{aligned}
\dot{V}_1 \le\ & \bar{e}^\top(\mathcal{P}\bar{\Lambda}_L + \bar{\Lambda}_L^\top \mathcal{P})\bar{e} - \rho_o\bar{e}^\top(\mathcal{P}\Pi + \Pi^T\mathcal{P})\bar{e} - 2\rho_o\tilde{\rho}_u\hat{\rho}_u - 2\rho_o\tilde{\rho}_\omega\hat{\rho}_\omega - 2\rho_o\tilde{\rho}_f\hat{\rho}_f - 2\rho_o\tilde{\rho}_\varepsilon\hat{\rho}_\varepsilon \\
& - 2\rho_o\tilde{\rho}_{fs}\hat{\rho}_{fs} - 2\rho_o Tr\left(\tilde{L}^\top\hat{L}\right) - 2\rho_o Tr\left(\tilde{W}^\top\hat{W}\right)
\end{aligned} \tag{D.19}$$

By applying Young's inequality from Lemma 1, the following inequalities are derived:

$$-2\rho_o\tilde{\rho}_u\hat{\rho}_u = -2\rho_o\tilde{\rho}_u(\tilde{\rho}_u + \rho_u) \le -\rho_o\tilde{\rho}_u^2 + \rho_o\rho_u^2 \tag{D.20}$$

$$-2\rho_o\tilde{\rho}_\omega\hat{\rho}_\omega = -2\rho_o\tilde{\rho}_\omega(\tilde{\rho}_\omega + \rho_\omega) \le -\rho_o\tilde{\rho}_\omega^2 + \rho_o\rho_\omega^2 \tag{D.21}$$

$$-2\rho_o\tilde{\rho}_f\hat{\rho}_f = -2\rho_o\tilde{\rho}_f(\tilde{\rho}_f + \rho_f) \le -\rho_o\tilde{\rho}_f^2 + \rho_o\rho_f^2 \tag{D.22}$$

$$-2\rho_o\tilde{\rho}_\varepsilon\hat{\rho}_\varepsilon = -2\rho_o\tilde{\rho}_\varepsilon(\tilde{\rho}_\varepsilon + \rho_\varepsilon) \le -\rho_o\tilde{\rho}_\varepsilon^2 + \rho_o\rho_\varepsilon^2 \tag{D.23}$$

$$-2\rho_o\tilde{\rho}_{fs}\hat{\rho}_{fs} = -2\rho_o\tilde{\rho}_{fs}\left(\tilde{\rho}_{fs}+\rho_{fs}\right) \le -\rho_o\tilde{\rho}_{fs}^2+\rho_o\rho_{fs}^2 \tag{D.24}$$

$$\begin{aligned}-2\rho_o Tr\left(\tilde{L}^\top\hat{L}\right) &= -2\rho_o Tr\left(\tilde{L}^\top\tilde{L}\right)+2\rho_o Tr\left(\tilde{L}^\top L\right) = -2\rho_o Tr\left(\tilde{L}^\top\tilde{L}\right)+2\rho_o\left(Vec\left(\tilde{L}\right)\right)^\top Vec(L)\\ &\le -2\rho_o Tr\left(\tilde{L}^\top\tilde{L}\right)+\rho_o\left(Vec\left(\tilde{L}\right)\right)^\top Vec\left(\tilde{L}\right)+\rho_o\left(Vec(L)\right)^\top Vec(L)\\ &= -2\rho_o Tr\left(\tilde{L}^\top\tilde{L}\right)+\rho_o Tr\left(\tilde{L}^\top\tilde{L}\right)+\rho_o Tr(L^\top L) = -\rho_o Tr\left(\tilde{L}^\top\tilde{L}\right)+\rho_o Tr(L^\top L)\end{aligned} \tag{D.25}$$

$$-2\rho_o Tr\left(\tilde{W}^\top\hat{W}\right) \le -\rho_o Tr\left(\tilde{W}^\top\tilde{W}\right)+\rho_o Tr(W^\top W) \tag{D.26}$$

Using Eq.(53) and (D.20)–(D.26), the expression in (D.19) can be simplified as follows:

$$\begin{aligned}\dot{V}_1 &\le \bar{e}^\top\mathcal{Q}\bar{e}-\rho_o\bar{e}^\top(P\Pi+\Pi^T P)\bar{e}-\rho_o\tilde{\rho}_u^2-\rho_o\tilde{\rho}_\omega^2-\rho_o\tilde{\rho}_f^2-\rho_o\tilde{\rho}_\varepsilon^2-\rho_o\tilde{\rho}_{fs}^2-\rho_o Tr\left(\tilde{L}^\top\tilde{L}\right)\\ &\quad-\rho_o Tr\left(\tilde{W}^\top\tilde{W}\right)\\ &\quad+\rho_o\left(\rho_u^2+\rho_\omega^2+\rho_f^2+\rho_\varepsilon^2+\rho_{fs}^2+Tr(L^\top L)+Tr(W^\top W)\right)\\ &\le -\rho_o\bar{e}^\top(P\Pi+\Pi^T P)\bar{e}-\rho_o\tilde{\rho}_u^2-\rho_o\tilde{\rho}_\omega^2-\rho_o\tilde{\rho}_f^2-\rho_o\tilde{\rho}_\varepsilon^2-\rho_o\tilde{\rho}_{fs}^2\\ &\quad-\rho_o Tr\left(\tilde{L}^\top\tilde{L}\right)-\rho_o Tr\left(\tilde{W}^\top\tilde{W}\right)\\ &\quad+\rho_o\left(\rho_u^2+\rho_\omega^2+\rho_f^2+\rho_\varepsilon^2+\rho_{fs}^2+Tr(L^\top L)+Tr(W^\top W)\right)\end{aligned} \tag{D.27}$$

By defining $\theta = \rho_u^2+\rho_\omega^2+\rho_f^2+\rho_\varepsilon^2+\rho_{fs}^2+Tr(L^\top L)+Tr(W^\top W)$ and $r = min\left\{\frac{\lambda_{min}(\mathcal{P}\Pi+\Pi^T\mathcal{P})}{\lambda_{max}(\mathcal{P})}, 2, \frac{1}{\lambda_{max}(\Gamma_L^{-1})}, \frac{1}{\lambda_{max}(\Gamma_W^{-1})}\right\}$,

Then,(D.27) can be expressed as:

$$\dot{V}_1 \le -\rho_o r V_1+\rho_o\theta \tag{D.28}$$

Appendix E. Proof of Theorem 2

Substituting Eq.(77) into Eq.(65) yields:

$$\begin{aligned}\dot{z} &= \mathcal{A}_m z+\rho_c\mathcal{A}_r z-\rho_c\mathcal{A}z+\mathcal{B}K_\Delta\Delta-\mathcal{B}\hat{\phi}(\hat{z}_a)+\mathcal{B}\Delta u-\mathcal{B}\tilde{u}+\mathcal{B}_\omega\omega(t)+\mathcal{B}_{fa}f_a(t)+\mathcal{B}_W\mathcal{W}(\xi,u_\tau)\\ &\quad+\breve{\mathcal{K}}e_\xi\end{aligned} \tag{E.1}$$

Based on the Neural Network defined in the observer design section, $\mathcal{W}(\xi,u_\tau) = W_W^T\Phi_W(\xi,u_\tau)+\varepsilon_W(\xi,u_\tau)$. Furthermore, from the DNN estimation error, $\hat{\phi}(\hat{z}_a) = \phi(z_a)-\tilde{\phi}_k$. Based on Section 3.2.2, $\tilde{\phi}_k = \sum_{j=0}^{\kappa}\Psi_j\mathrm{Vec}\left(\tilde{\vartheta}_j\right)+\sum_{j=1}^{k}\Xi_{\phi_j}\Delta_{\phi_j}$ with $\Psi_j = \Xi_{\xi_j}\left(I_{l_{j+1}}\otimes\hat{\psi}_j^\top\right)$, $\Psi_0 = \Xi_{\xi_0}\left(I_{l_1}\otimes z_a^T\right)$, $\Xi_{\xi_j} = \prod_{m=j+1}^{\breve{\kappa}}\vartheta_m^\top\hat{\psi}_m' = \vartheta_\kappa^\top\hat{\psi}_\kappa'\vartheta_{\kappa-1}^\top\hat{\psi}_{\kappa-1}'\ldots\vartheta_{j+1}^\top\hat{\psi}_{j+1}'$.

Thus,

$$\begin{aligned}\dot{z} &= \mathcal{A}_m z+\rho_c\mathcal{A}_r z+\breve{\mathcal{K}}e_\xi+\mathcal{B}K_\Delta\Delta-\mathcal{B}\phi(z_a)+\mathcal{B}\tilde{\phi}_k+\mathcal{B}\Delta u\ -\mathcal{B}\tilde{u}+\mathcal{B}_\omega\omega(t)+\mathcal{B}_{fa}f_a(t)\\ &\quad+\mathcal{B}_W W_W^T\Phi_W(\xi,u_\tau)+\mathcal{B}_W\varepsilon_W(\xi,u_\tau)-\rho_c\mathcal{A}z\\ &= \mathcal{A}_m z+\rho_c\mathcal{A}_r z+\breve{\mathcal{K}}e_\xi+\mathcal{B}K_\Delta\Delta-\mathcal{B}\phi(z_a)+\mathcal{B}\sum_{j=0}^{\kappa}\Psi_j\mathrm{Vec}\left(\tilde{\vartheta}_j\right)+\mathcal{B}\sum_{j=1}^{k}\Xi_{\phi_j}\Delta_{\phi_j}\\ &\quad+\mathcal{B}\Delta u-\mathcal{B}\tilde{u}+\mathcal{B}_\omega\omega(t)+\mathcal{B}_{fa}f_a(t)+\mathcal{B}_W W_W^T\Phi_W(\xi,u_\tau)+\mathcal{B}_W\varepsilon_W(\xi,u_\tau)-\rho_c\mathcal{A}z\end{aligned} \tag{E.2}$$

Let the Lyapunov function be defined as:

$$V_2 = \frac{1}{2}z^T Pz+\frac{1}{2}\Delta^T\Gamma_\Delta^{-1}\Delta+\frac{1}{2}\sum_{j=0}^{\kappa}Vec\left(\tilde{\vartheta}_j\right)^T\Gamma_{\vartheta j}^{-1}Vec\left(\tilde{\vartheta}_j\right) \tag{E.3}$$

With $\tilde{\vartheta}_j = \vartheta_j-\hat{\vartheta}_j$. The time derivative of (E.3) is:

$$\dot{V}_2 = \frac{1}{2}z^T P\dot{z} + \frac{1}{2}\dot{z}^T Pz + \frac{1}{2}\Delta^T \Gamma_\Delta^{-1}\dot{\Delta} + \frac{1}{2}\dot{\Delta}^T \Gamma_\Delta^{-1}\Delta + \frac{1}{2}\sum_{j=0}^{\kappa} Vec(\tilde{\vartheta}_j)^T \Gamma_{\vartheta j}^{-1} Vec\left(\dot{\tilde{\vartheta}}_j\right)$$
$$+ \frac{1}{2}\sum_{j=0}^{\kappa} Vec\left(\dot{\tilde{\vartheta}}_j\right)^T \Gamma_{\vartheta j}^{-1} Vec(\tilde{\vartheta}_j) \tag{E.4}$$
$$= \frac{1}{2}z^T P\dot{z} + \frac{1}{2}\dot{z}^T Pz + \Delta^T \Gamma_\Delta^{-1}\dot{\Delta} - \sum_{j=0}^{\kappa} Vec(\tilde{\vartheta}_j)^T \Gamma_{\vartheta j}^{-1} Vec\left(\dot{\tilde{\vartheta}}_j\right)$$

Substituting (E.2) into (E.4) results in

$$\dot{V}_2 = \frac{1}{2}z^T(\mathcal{A}_m^T P + P\mathcal{A}_m)z + \frac{1}{2}\rho z^T(\mathcal{A}_r^T P + P\mathcal{A}_r - 2P\mathcal{A})z + z^T P\breve{\mathcal{K}}e_\xi + z^T P\mathcal{B}K_\Delta\Delta - z^T P\mathcal{B}\phi(z_a)$$
$$+ z^T P\mathcal{B}\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) + z^T P\mathcal{B}\sum_{j=1}^{\kappa}\Xi_{\phi_j}\Delta_{\phi_j} + z^T P\mathcal{B}\Delta u - z^T P\mathcal{B}\tilde{u} + z^T P\mathcal{B}_\omega\omega(t)$$
$$+ z^T P\mathcal{B}_{fa}f_a(t) + z^T P\mathcal{B}_W W_W^T\Phi_W(\xi,u_\tau) + z^T P\mathcal{B}_W\varepsilon_W(\xi,u_\tau) + \Delta^T\Gamma_\Delta^{-1}\dot{\Delta} \tag{E.5}$$
$$- \sum_{j=0}^{\kappa} Vec(\tilde{\vartheta}_j)^T \Gamma_{\vartheta j}^{-1} Vec\left(\dot{\tilde{\vartheta}}_j\right)$$

By defining $e_{fs} = \hat{\xi}_{fs} - \xi_{fs}$ and using Eq.(79), we have:

$$z^T P\mathcal{B}\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) = \hat{z}^T P\mathcal{B}\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) - e_\xi^T\mathcal{K}^T P\mathcal{B}\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) =$$
$$= \hat{z}^T P\mathcal{B}\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) + e_y^T\mathcal{M}^T\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) + e_{fs}^T\mathcal{M}^T\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) \tag{E.6}$$

The application of (E.6) to (E.5) leads to:

$$\dot{V}_2 = \frac{1}{2}z^T(\mathcal{A}_m^T P + P\mathcal{A}_m)z + \frac{1}{2}\rho z^T(\mathcal{A}_r^T P + P\mathcal{A}_r - 2P\mathcal{A})z + z^T P\breve{\mathcal{K}}e_\xi + z^T P\mathcal{B}K_\Delta\Delta - z^T P\mathcal{B}\phi(z_a)$$
$$+ \hat{z}^T P\mathcal{B}\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) + e_y^T\mathcal{M}^T\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) + e_{fs}^T\mathcal{M}^T\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j)$$
$$+ z^T P\mathcal{B}\sum_{j=1}^{\kappa}\Xi_{\phi_j}\Delta_{\phi_j} + z^T P\mathcal{B}\Delta u - z^T P\mathcal{B}\tilde{u} + z^T P\mathcal{B}_\omega\omega(t) + z^T P\mathcal{B}_{fa}f_a(t) \tag{E.7}$$
$$+ z^T P\mathcal{B}_W W_W^T\Phi_W(\xi,u_\tau) + z^T P\mathcal{B}_W\varepsilon_W(\xi,u_\tau) + \Delta^T\Gamma_\Delta^{-1}\dot{\Delta} - \sum_{j=0}^{\kappa} Vec(\tilde{\vartheta}_j)^T \Gamma_{\vartheta j}^{-1} Vec\left(\dot{\tilde{\vartheta}}_j\right)$$

Employing Young's inequality gives

$$z^T P\breve{\mathcal{K}}e_\xi = z^T P\mathcal{B}K\mathcal{K}e_\xi + \rho z^T P\mathcal{B}K_r\mathcal{K}e_\xi$$
$$\leq \alpha_1 z^T z + \frac{1}{\alpha_1}e_\xi^T\mathcal{K}^T K^T\mathcal{B}^T PP\mathcal{B}K\mathcal{K}e_\xi + \rho\alpha_2 z^T z + \frac{\rho}{\alpha_2}e_\xi^T\mathcal{K}^T K_r^T\mathcal{B}^T PP\mathcal{B}K_r\mathcal{K}e_\xi \tag{E.8}$$

Since the observer estimation error e_ξ is bounded, there exists positive constants β_1 and β_2 ensuring that the inequality $\frac{1}{\alpha_1}e_\xi^T\mathcal{K}^T K^T\mathcal{B}^T PP\mathcal{B}K\mathcal{K}e_\xi \leq \beta_1$ and $\frac{1}{\alpha_2}e_\xi^T\mathcal{K}^T K_r^T\mathcal{B}^T PP\mathcal{B}K_r\mathcal{K}e_\xi \leq \beta_2$ holds. Therefore,

$$z^T P\breve{\mathcal{K}}e_\xi \leq \alpha_1 z^T z + \beta_1 + \rho\alpha_2 z^T z + \rho\beta_2 \tag{E.9}$$

With $\beta_1 = \frac{1}{\alpha_1}\lambda_{\max}(\mathcal{K}^T K^T\mathcal{B}^T PP\mathcal{B}K\mathcal{K})\bar{e}_\xi$ and $\beta_2 = \frac{1}{\alpha_2}\lambda_{\max}(\mathcal{K}^T K_r^T\mathcal{B}^T PP\mathcal{B}K_r\mathcal{K})\bar{e}_\xi$, $\|e_\xi\|^2 \leq \bar{e}_\xi$. The other terms in (E.7) satisfy the following inequalities:

$$z^T P\mathcal{B}K_\Delta\Delta \leq \alpha_3 z^T z + \Delta^T\beta_3\Delta \tag{E.10}$$

With $\beta_3 = \frac{1}{\alpha_3}K_\Delta^T\mathcal{B}^T PP\mathcal{B}K_\Delta$. Given the boundedness of the Deep Neural Network $\phi(z_a)$, as discussed in Section 3.2.2, it follows that:

$$-z^T P\mathcal{B}\phi(z_a) \le \alpha_4 z^T z + \beta_4 \tag{E.11}$$

Where $\beta_4 = \frac{1}{\alpha_4}\lambda_{max}(\mathcal{B}^T PP\mathcal{B})\bar{\phi}$, $\|\phi(z_a)\|^2 \le \bar{\phi}$. Expanding $e_{fs}^T\mathcal{M}^T\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j)$ results in:

$$\begin{aligned} e_{fs}^T\mathcal{M}^T\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) &= e_{fs}^T\mathcal{M}^T\left(\Psi_0 Vec(\tilde{\vartheta}_0) + \Psi_1 Vec(\tilde{\vartheta}_1) + \cdots + \Psi_\kappa Vec(\tilde{\vartheta}_\kappa)\right) \\ &= Vec^T(\tilde{\vartheta}_0)\Psi_0^T\mathcal{M}e_{fs} + Vec^T(\tilde{\vartheta}_1)\Psi_1^T\mathcal{M}e_{fs} + \cdots + Vec^T(\tilde{\vartheta}_\kappa)\Psi_\kappa^T\mathcal{M}e_{fs} \end{aligned} \tag{E.12}$$

The boundedness of $\Psi_j^T\mathcal{M}e_{fs}$ is ensured by the boundedness of Ψ_j, $\mathcal{M}$, and e_{fs}. A subsequent application of Young's inequality yields:

$$e_{fs}^T\mathcal{M}\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) \le \alpha_5\sum_{j=0}^{\kappa}Vec(\tilde{\vartheta}_j)^T Vec(\tilde{\vartheta}_j) + \beta_5 \tag{E.13}$$

Given the boundedness of $\sum_{j=1}^{\kappa}\Xi_{\xi_j}\Delta_{\xi_j}$, e_{fs}, $\tilde{u}$, $\omega(t)$ and $f_a(t)$ and employing of Young's inequalities one has:

$$z^T P\mathcal{B}\sum_{j=1}^{\kappa}\Xi_{\xi_j}\Delta_{\xi_j} \le \alpha_5 z^T z + \frac{1}{\alpha_5}\left(\sum_{j=1}^{\kappa}\Xi_{\xi_j}\Delta_{\xi_j}\right)^T\mathcal{B}^T PP\mathcal{B}\sum_{j=1}^{\kappa}\Xi_{\xi_j}\Delta_{\xi_j} \le \alpha_6 z^T z + \beta_6 \tag{E.14}$$

$$\begin{aligned} z^T P\mathcal{B}\Delta u = \hat{z}^T P\mathcal{B}\Delta u - e_\xi^T\mathcal{K}^T P\mathcal{B}\Delta u &= \hat{z}^T P\mathcal{B}\Delta u + e_y^T\mathcal{M}^T\Delta u + e_{fs}^T\mathcal{M}^T\Delta u \\ &\le \|\hat{z}^T P\mathcal{B}\Delta u\| + \left\|e_y^T\mathcal{M}^T\Delta u\right\| + \alpha_7\|\mathcal{M}^T\Delta u\|^2 + \frac{1}{\alpha_7}\left\|e_{fs}\right\|^2 \\ &\le \|\hat{z}^T P\mathcal{B}\Delta u\| + \left\|e_y^T\mathcal{M}^T\Delta u\right\| + \alpha_7\|\mathcal{M}^T\Delta u\|^2 + \beta_7 \end{aligned} \tag{E.15}$$

In a similar way we have

$$-z^T P\mathcal{B}\tilde{u} \le \alpha_8 z^T z + \frac{1}{\alpha_8}\tilde{u}^T\mathcal{B}^T PP\mathcal{B}\tilde{u} \le \alpha_8 z^T z + \beta_8 \tag{E.16}$$

$$z^T P\mathcal{B}\omega(t) \le \alpha_9 z^T z + \frac{1}{\alpha_9}\omega^T\mathcal{B}^T PP\mathcal{B}\omega \le \alpha_9 z^T z + \beta_9 \tag{E.17}$$

$$z^T P\mathcal{B}f_a(t) \le \alpha_{10} z^T z + \frac{1}{\alpha_{10}}f_a^T\mathcal{B}^T PP\mathcal{B}f_a \le \alpha_{10} z^T z + \beta_{10} \tag{E.18}$$

Furthermore, the property of the Neural Network leads to: $\|W_{\mathcal{W}}\| \le \bar{W}_{\mathcal{W}}$, $\|\Phi_{\mathcal{W}}(\xi,u_\tau)\| \le \bar{\Phi}_{\mathcal{W}}$, $\|\varepsilon_{\mathcal{W}}(\xi,u_\tau)\|^2 \le \bar{\varepsilon}_{\mathcal{W}}$, $\|W_{\mathcal{W}}^T\Phi_{\mathcal{W}}(\xi,u_\tau)\|^2 \le c_2$, hence, $z^T P\mathcal{B}_{\mathcal{W}}W_{\mathcal{W}}^T\Phi_{\mathcal{W}}(\xi,u_\tau) + z^T P\mathcal{B}_{\mathcal{W}}\varepsilon_{\mathcal{W}}(\xi,u_\tau)$

$$z^T P\mathcal{B}_{\mathcal{W}}W_{\mathcal{W}}^T\Phi_{\mathcal{W}}(\xi,u_\tau) \le \alpha_{11} z^T z + \frac{1}{\alpha_{11}}\Phi_{\mathcal{W}}^T(\xi,u_\tau)W_{\mathcal{W}}\mathcal{B}_{\mathcal{W}}^T PP\mathcal{B}_{\mathcal{W}}W_{\varpi}^T\Phi_{\mathcal{W}}(\xi,u_\tau) \le \alpha_{11} z^T z + \beta_{11} \tag{E.19}$$

$$z^T P\mathcal{B}_{\mathcal{W}}\varepsilon_{\mathcal{W}}(\xi,u_\tau) \le \alpha_{12} z^T z + \frac{1}{\alpha_{12}}\varepsilon_{\mathcal{W}}^T\mathcal{B}_{\mathcal{W}}^T PP\mathcal{B}_{\mathcal{W}}\varepsilon_{\mathcal{W}} \le \alpha_{12} z^T z + \beta_{12} \tag{E.20}$$

Thus (E.7) becomes:

$$\begin{aligned} \dot{V}_2 \le{}& \frac{1}{2}z^T(\mathcal{A}_m^T P + P\mathcal{A}_m)z + \frac{1}{2}\rho z^T(\mathcal{A}_r^T P + P\mathcal{A}_r - 2P\mathcal{A})z + \alpha_1 z^T z + \beta_1 + \rho\alpha_2 z^T z + \rho\beta_2 + \alpha_3 z^T z \\ &+ \Delta^T\beta_3\Delta + \alpha_4 z^T z + \beta_4 + \hat{z}^T P\mathcal{B}\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) + e_y^T\mathcal{M}^T\sum_{j=0}^{\kappa}\Psi_j Vec(\tilde{\vartheta}_j) \\ &+ \alpha_5\sum_{j=0}^{\kappa}Vec(\tilde{\vartheta}_j)^T Vec(\tilde{\vartheta}_j) + \beta_5 + \alpha_6 z^T z + \beta_6 + \|\hat{z}^T P\mathcal{B}\Delta u\| + \left\|e_y^T\mathcal{M}^T\Delta u\right\| \\ &+ \alpha_7\|\mathcal{M}^T\Delta u\|^2 + \beta_7 + \alpha_8 z^T z + \beta_8 + \alpha_9 z^T z + \beta_9 + \alpha_{10} z^T z + \beta_{10} + \alpha_{11} z^T z + \beta_{11} \\ &+ \alpha_{12} z^T z + \beta_{12} + \Delta^T\Gamma_\Delta^{-1}\dot{\Delta} - \sum_{j=0}^{\kappa}Vec(\tilde{\vartheta}_j)^T\Gamma_{\vartheta j}^{-1}Vec\left(\dot{\tilde{\vartheta}}_j\right) \end{aligned} \tag{E.21}$$

Upon substituting equations Eq.(77) and Eq.(78) into (E.21), the result is:

$$\dot{V}_2 \le \frac{1}{2}z^T(\mathcal{A}_m^T P + P\mathcal{A}_m + \alpha_s)z + \frac{1}{2}\rho z^T(\mathcal{A}_r^T P + P\mathcal{A}_r - 2P\mathcal{A} + \alpha_t)z + \beta_s + \rho\beta_t + \Delta^T\beta_3\Delta + \alpha_4\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\tilde{\vartheta}_j) - \Delta^T\sigma_\Delta\Delta - \rho\Delta^T\Delta + \sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\sigma_{\vartheta j}\mathrm{Vec}(\hat{\vartheta}_j) + \rho\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\hat{\vartheta}_j) \tag{E.22}$$

where $\alpha_s = 2(\alpha_1 + \alpha_3 + \alpha_4 + \alpha_6 + \alpha_8 + \alpha_9 + \alpha_{10} + \alpha_{11} + \alpha_{12})$, $\alpha_t = 2\alpha_2$, $\beta_s = \beta_1 + \beta_4 + \beta_5 + \beta_6 + \beta_7 + \beta_8 + \beta_9 + \beta_{10} + \beta_{11} + \beta_{12}$, $\beta_t = \beta_2$. α_j and β_j, $j = 1,2,\dots,12$ are positive constants.

Based on the definition of $\tilde{\vartheta}_j$ and applying Young's inequality, we obtain:

$$\begin{aligned}\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\sigma_{\vartheta j}\mathrm{Vec}(\hat{\vartheta}_j) &= -\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\sigma_{\vartheta j}\mathrm{Vec}(\tilde{\vartheta}_j) + \sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\sigma_{\vartheta j}\mathrm{Vec}(\vartheta_j)\\ &\le -\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\sigma_{\vartheta j}\mathrm{Vec}(\tilde{\vartheta}_j) + \alpha_{13}\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\tilde{\vartheta}_j)\\ &+ \frac{1}{\alpha_{13}}\sum_{j=0}^{\kappa}\mathrm{Vec}(\vartheta_j)^T\sigma_j^T\sigma_{\vartheta j}\mathrm{Vec}(\vartheta_j)\\ &= -\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\left(\sigma_{\vartheta j} - \alpha_{13}I_{l_j l_{j+1}}\right)\mathrm{Vec}(\tilde{\vartheta}_j) + \beta_{13}\end{aligned} \tag{E.23}$$

Similarly,

$$\begin{aligned}\rho\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\hat{\vartheta}_j) &= -\rho\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\tilde{\vartheta}_j) + \rho\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\vartheta_j)\\ &\le -\rho\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\tilde{\vartheta}_j) + \alpha_{14}\rho\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\tilde{\vartheta}_j)\\ &+ \frac{1}{\alpha_{14}}\rho\sum_{j=0}^{\kappa}\mathrm{Vec}(\vartheta_j)^T\sigma_{\vartheta j}\mathrm{Vec}(\vartheta_j) = -\rho(1-\alpha_{14})\sum_{j=0}^{\kappa}\mathrm{Vec}(\tilde{\vartheta}_j)^T\mathrm{Vec}(\tilde{\vartheta}_j) + \rho\beta_{14}\end{aligned} \tag{E.24}$$

Where $\beta_{14} = \frac{1}{\alpha_{14}}\sum_{j=0}^{\kappa}\mathrm{Vec}(\vartheta_j)^T\sigma_{\vartheta j}\mathrm{Vec}(\vartheta_j)$. Applying Eq.(80), Eq.(81), (E.23), and (E.24) into (E.22) yields:

$$\begin{aligned}\dot{V}_2 \le{}& \frac{1}{2}z^T Q_1 z + \frac{1}{2}\rho z^T Q_2 z - \Delta^T(\sigma_\Delta - \beta_3)\Delta - \rho\Delta^T\Delta\\ &- \sum_{j=0}^{\kappa}Vec(\tilde{\vartheta}_j)^T\left(\sigma_{\vartheta j} - (\alpha_5 + \alpha_{13})I_{l_j l_{j+1}}\right)Vec(\tilde{\vartheta}_j)\\ &- \rho(1-\alpha_{14})\sum_{j=0}^{\kappa}Vec(\tilde{\vartheta}_j)^T Vec(\tilde{\vartheta}_j) + \beta_{13} + \beta_s + \rho\beta_{14} + \rho\beta_t\end{aligned} \tag{E.25}$$

If Eq.(82)-Eq.(84) hold and defining $\breve{\beta} = \beta_{14} + \beta_s$, $\breve{\beta} = \beta_{15} + \beta_t$, $\breve{\alpha} = 1 - \alpha_{14}$ and $Q = -Q_2$, then:

$$\dot{V}_2 \le -\frac{1}{2}\rho z^T Q z - \rho\Delta^T\Delta - \rho\breve{\alpha}\sum_{j=0}^{\kappa}Vec(\tilde{\vartheta}_j)^T Vec(\tilde{\vartheta}_j) + \breve{\beta} + \rho\breve{\beta} \tag{E.26}$$

Consider $\mathfrak{T} = min\left\{\frac{\lambda_{min}(Q)}{\lambda_{max}(P)}, \frac{2}{\lambda_{max}(\Gamma_\Delta^{-1})}, \frac{2\breve{\alpha}}{\lambda_{max}(\Gamma_{\vartheta j}^{-1})}\right\}$.

Thus, the following inequality holds:

$$\dot{V}_2 \le -\rho\mathfrak{T}V_2 + \breve{\beta} + \rho\breve{\beta} \tag{E.27}$$

Based on Lemma 3, the state variables z of the continuum robot system exhibit prescribed-time input-to-state stability (PT-ISS) and converge to an adjustable neighborhood of the origin, as shown by $\|z\| \leq \sqrt{\frac{\mathcal{C}}{\mathfrak{T}\lambda_{min}(P)}}$, $t \geq T_p$, where $\mathcal{C} = \check{\beta} + \rho\check{\beta}$. This completes the proof of Theorem 2 ■.

Appendix F: Parameters of the Observer (Theorem 1) and the Controller (Theorem 2)

The unknown matrices in Theorems 1–2 were solved using YALMIP for the corresponding matrix equalities and inequalities. The numerical values of all relevant parameters and matrices used in the simulations are given as follows.

$$\bar{\Lambda}_L = \begin{bmatrix} -28.92498 & 2.973302e-06 & 1 & 0 & -28.92498 & 2.973302e-06 \\ 1.260619e-06 & -28.925106 & 0 & 1 & 1.260619e-06 & -28.92510 \\ -49.60722 & 5.948136e-06 & 0 & 0 & -49.60722 & 5.948136e-06 \\ 2.475317e-06 & -49.60743 & 0 & 0 & 2.475317e-06 & -49.607435 \\ 20.94359 & -3.356494e-06 & 0 & 0 & 19.94359 & -3.356494e-06 \\ -1.847612e-06 & 20.94370 & 0 & 0 & -1.847612e-06 & 19.943701 \end{bmatrix}$$

$$\mathcal{P} = \begin{bmatrix} 0.06050 & -5.9183e-09 & -0.0073 & -7.3457e-10 & 0.0644 & -5.4808e-09 \\ -5.9183e-09 & 0.06050 & 2.4842e-10 & -0.00732 & -6.7183e-09 & 0.0644 \\ -0.0073 & 2.4842e-10 & 0.0026 & 9.3284e-11 & -0.0070 & 2.0313e-10 \\ -7.3457e-10 & -0.0073 & 9.32843e-11 & 0.0026 & -7.1562e-10 & -0.0070 \\ 0.0644 & -6.7183e-09 & -0.00707 & -7.1562e-10 & 0.0739 & -6.2569e-09 \\ -5.4808e-09 & 0.0644 & 2.0313e-10 & -0.00707 & -6.2569e-09 & 0.0739 \end{bmatrix}$$

$$\mathcal{Q}_1 = \mathcal{Q}_2 = \mathcal{Q}_3 = \begin{bmatrix} -0.0071 & 3.1648e-10 \\ -9.29168e-10 & -0.0071 \end{bmatrix}, \mathcal{Q}_4 = \begin{bmatrix} 4.42444e-10 & -1.37571e-10 \\ 4.5003e-10 & -5.8937e-11 \end{bmatrix}$$

$$\mathcal{Q}_\varepsilon = \begin{bmatrix} -0.0067 & 8.2307e-10 \\ -4.5187e-10 & -0.0067 \\ -0.0071 & 3.10417e-10 \\ -8.4515e-10 & -0.00719 \\ 2.4456e-10 & 2.2226e-10 \\ -5.0993e-10 & -2.5494e-10 \end{bmatrix}, \mathcal{Q}_3 = \begin{bmatrix} -0.0067 & 8.23076e-10 \\ -4.5187e-10 & -0.00673 \\ -0.00719 & 3.10417e-10 \\ -8.4515e-10 & -0.00719 \\ -673213 & 10.45337 \\ -9.61811 & -673214 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.03245 & -1.0268e-07 & 0.0062 & 1.3765 & 0.0163 & 2.9170e-08 \\ -1.02682e-07 & 0.0324 & 1.3765e-07 & 0.0062 & 2.91717e-08 & 0.0163 \\ 0.0062 & 1.3765e-07 & 0.0064 & 6.7069 & 0.0040 & 9.3573e-08 \\ 1.3765e-07 & 0.0062 & 6.7069e-08 & 0.0064 & 9.3573e-08 & 0.0040 \\ 0.0163 & 2.9171e-08 & 0.0040 & 9.3573e-08 & 0.0217 & 3.5619e-08 \\ 2.9170e-08 & 0.0163 & 9.3573e-08 & 0.00403 & 3.5619e-08 & 0.0217 \end{bmatrix}$$

$$\mathcal{A}_m = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -12435 & -1133 & -197.6337 & -8.7240 & -245777 & -36092 \\ -1018 & -12117 & -8.2486 & -192.3662 & -28622 & -250279 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{A}_r = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -6.3333 & -5.4499e-16 & -4.76094 & 2.6892e-16 & -3.1622 & -3.5162e-16 \\ -7.3949e-17 & -6.3333 & -7.7151e-17 & -4.7609 & 6.7525e-18 & -3.1622 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{K} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathcal{M} = \begin{bmatrix} -0.3373 & -0.3444 \\ -0.3444 & -0.3381 \end{bmatrix}$$

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Mohammad Jabari reports financial support was provided by PNRR-NGEU. This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 351/2022.If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.