Self-attention Enhanced Dynamics Learning and Adaptive Fractional-order Control for Continuum Soft Robots with System Uncertainties

Shao, Xiangyu; Xu, Linke; Sun, Guanghui; Yao, Weiran; Wu, Ligang; Santina, Cosimo Della

DOI
10.1109/TASE.2025.3590174

Publication date
2025

Document Version
Final published version

Published in
IEEE Transactions on Automation Science and Engineering

Self-Attention Enhanced Dynamics Learning and Adaptive Fractional-Order Control for Continuum Soft Robots With System Uncertainties

Xiangyu Shao, *Member, IEEE*, Linke Xu, *Graduate Student Member, IEEE*, Guanghui Sun, *Senior Member, IEEE*, Weiran Yao, *Member, IEEE*, Ligang Wu, *Fellow, IEEE*, and Cosimo Della Santina, *Senior Member, IEEE*

***Abstract*—Dynamics-based control offers a promising approach to exploring the motion potential of soft robots. However, inherently infinite degrees of freedom of these systems pose significant challenges for dynamics modeling, closely followed by the pressing robustness concerns arising from finite-dimensional approximations. This paper addresses these issues by proposing a physics-informed dynamics learning neural network and an adaptive fractional-order control for continuum soft robots. Specifically, a deep Lagrangian neural network is first developed with an embedded self-attention mechanism to enhance learning efficiency, accuracy, and data sensitivity. Subsequently, an adaptive fractional-order sliding mode controller is designed, leveraging the inherent historical memory properties of fractional calculus. This controller not only ensures robust shape control but also improves response speed and tracking accuracy. To further handle model discrepancies in the learned dynamics and external disturbances, a nonlinear disturbance observer is introduced to effectively estimate and compensate for lumped uncertainties, thereby ensuring reliable performance. Theoretical analysis confirms the closed-loop stability, while both simulation and experiment results validate the high dynamics fitting accuracy of the proposed network, as well as the robust and precise tracking capability of the fractional-order controller.**

***Note to Practitioners*—Soft robots offer great potential in unstructured or constrained environments owing to their compliance and adaptability. However, their high degrees of freedom and nonlinear behaviors make analytical modeling and robust control particularly challenging. Meanwhile, traditional closed-box learning methods often suffer from limited physical interpretability, reliability and extrapolability. This work presents a physics-informed dynamics learning framework combined with a fractional-order controller for soft robots. The dynamics learning network embeds physical priors to enhance model interpretability and extrapolability, while a self-attention mechanism improves data efficiency and modeling accuracy. Additionally, a disturbance observer is designed to estimate and compensate for model discrepancies and external disturbances, thereby contributing to the system's robustness. Incorporating the observer's outputs, the adaptive fractional-order controller further enhances closed-loop behavior by leveraging the memory properties of fractional calculus.**

***Index Terms*—Continuum soft robot, physics-informed neural network, fractional-order sliding mode, self-attention.**

I. Introduction

CONTINUUM soft robots are composed almost entirely of soft elastic materials, exhibiting great potential in tasks involving human interaction or navigation through confined, obstacle-laden environments, owing to their inherent flexibility and high degrees of freedom [1]. However, the intrinsic deformability and nonlinearity of soft materials that makes these tasks possible also impose significant challenges for dynamics modeling and controller design [2]. As a result, developing practical modeling and control framework is essential to advancing the motor intelligence of soft robots.

Over the past decades, numerous researchers have devoted significant efforts to addressing these challenges. Regarding the kinematics modeling, Webster and Jones [3] proposed a piecewise constant curvature (PCC) approach, where the soft robot is segmented into multiple smoothly connected arcs, each characterized by identical parameters. Mahl et al. [4] refined the PCC assumption by introducing a variable curvature hypothesis, providing a more accurate representation of the robot's motion. Renda et al. [5] investigated the piecewise constant strain hypothesis based on Cosserat rod theory, modeling the soft robot as a sequence of elastic rod deformations.

Received 9 January 2025; revised 7 May 2025; accepted 11 July 2025. Date of publication 17 July 2025; date of current version 25 July 2025. This article was recommended for publication by Associate Editor H.-T. Yau and A. M. Ghalamzan Esfahani upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62303137 and Grant 62173107, in part by the Special funding of China Postdoctoral Science Foundation under Grant 2023TQ0092, in part by the Natural Science Foundation of Sichuan Province under Grant 2025ZNSFSC0461, in part by the New Era Heilongjiang Outstanding Master's and Doctoral Dissertation Foundation under Grant LJYXL2023-029, in part by Heilongjiang Postdoctoral Science Foundation under Grant LBH-Z20057, and in part by the Pre-research Task of State Key Laboratory of Robotics and System (HIT) under Grant SKLRS202404B. *(Corresponding author: Guanghui Sun.)*

Xiangyu Shao, Guanghui Sun, Weiran Yao, and Ligang Wu are with the State Key Laboratory of Robotics and System and the School of Astronautics, Harbin Institute of Technology (HIT), Harbin 150001, China, and also with Suzhou Research Institute of HIT, Suzhou 215104, China (e-mail: xiangyushao@hit.edu.cn; guanghuisun@hit.edu.cn; yaoweiran@hit.edu.cn; ligangwu@hit.edu.cn).

Linke Xu is with the State Key Laboratory of Robotics and System and the School of Astronautics, HIT, Harbin 150001, China (e-mail: linkexu@stu.hit.edu.cn).

Cosimo Della Santina is with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands, and also with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Weßling, Germany, and also with the Department of Informatics, Technical University of Munich, 80333 Munich, Germany (e-mail: C.DellaSantina@tudelft.nl).

This article has supplementary downloadable material available at https://doi.org/10.1109/TASE.2025.3590174, provided by the authors.

Digital Object Identifier 10.1109/TASE.2025.3590174

For dynamics modeling of soft robots, incorporating kinematic assumptions into Lagrangian or Hamiltonian equations is a common practice due to their physical interpretability and relative simplicity. However, such approaches require accurate knowledge of the robot's shape and material properties, resulting in limited modeling accuracy and generalizability. Deep learning methods utilize experimental data to capture unmodeled dynamics, avoiding complex analytical derivations and showing promise in soft robot modeling [6]. Navez et al. [7] proposed a condensed finite element model training framework to learn a unified representation across different soft robot designs, serving as a general basis for modeling, control, and design of soft manipulators. Sharma and Kramer [8] developed a physics-preserving and nonintrusive method that enables stable and accurate learning of reduced-order Lagrangian models solely from high-dimensional trajectory data. Lawal et al. [9] established principles of physics-informed neural networks (PINNs), which were further developed into deep Lagrangian neural networks (DeLaN) [10] and Hamiltonian neural networks [11]. By incorporating physical information into the network architecture, these methods achieve enhanced learning efficiency, model generalization, and physical consistency.

Among the methods discussed above, PINN-based dynamics learning is particularly noteworthy for its minimal reliance on prior system knowledge and robust extrapolation capabilities. These attributes make it particularly suitable for modeling complex continuum soft robots. Despite its promising potential, current PINN algorithms face challenges in balancing modeling accuracy and computational cost. Fitting dynamic parameters accurately poses a challenge, even when constrained to a limited number of training epochs. However, increasing the number of epochs could lead to overfitting, which in turn may diminish the model's ability to generalize effectively. Inspired by Huang et al. [12], incorporating the self-attention mechanism into PINNs offers a solution to these issues. It achieves this by decreasing the model's parameter count and reducing computational demands through efficient parameter sharing. Additionally, it improves the network's feature extraction capabilities, overall flexibility and scalability. As a result, this approach can significantly boost both the efficiency and effectiveness of PINNs in dynamics modeling.

Control strategies of continuum soft robots can be broadly classified as model-based or data-driven. Model-based approaches include modified PID control [13], sliding mode control (SMC) [14], [15] and fuzzy logic control [16], [17], whereas data-driven methods encompass iterative learning control [18] and neural network-based control [19], [20]. Among these, SMCs stand out for their simple, intuitive design and robustness against disturbance. However, conventional integer-order SMCs often fall short when tasked with governing the deformable, viscoelastic dynamics. As a generalization of integer-order derivatives endowed with intrinsic memory effects, fractional-order calculus offers a powerful framework for characterizing physical processes with historical dependence, such as flexible structures and biomedical systems [21]. Beyond its memory capacity, fractional calculus excels at modeling nonlinear, non-stationary, and non-Markovian dynamics. Embedding fractional derivatives into the SMC paradigm can therefore substantially improve the controller's adaptability and performance in handling the flexibility and deformability of soft robots. Recent works on fractional-order SMCs for soft robots [22], [23] have empirically validated these benefits.

Inspired by the robust extrapolation and physical interpretability of PINNs, this paper introduces a self-attention-enhanced dynamics learning neural network, an adaptive fractional-order sliding mode controller built upon the learned model, and an integrated nonlinear disturbance observer (NDOB) to compensate for lumped uncertainties. The principal contributions and connection to existing research are summarized as follows:

1) Whereas prior DeLaN variants [24], [25] treat every input equally, the proposed method embeds a self-attention mechanism into DeLaN. By dynamically weighting informative and representative samples while downplaying less relevant ones, it boosts learning efficiency, strengthens extrapolation, and enhances the physical interpretability of the inferred dynamics.
2) Compared to the prevailing—and predominantly used-integer-order controllers, this paper proposes an adaptive fractional-order SMC for soft robots. By harnessing the memory properties of fractional-order calculus, the controller effectively suppresses high-frequency disturbances and reduces chattering.
3) This paper embeds data-driven insights into the model-based controller design, with trajectory tracking experiments on a pneumatically actuated soft robots being conducted to verify superior performance of the proposed scheme.

The rest of this paper is organized as follows: Section II presents the design of self-attention enhanced physics informed Lagrange neural network. The adaptive fractional-order shape controller and nonlinear disturbance observer are proposed in Section III. Simulation and experiment results are given in Section IV and Section V, and Section VI concludes this paper.

II. Dynamics Learning via Self-Attention Enhanced Deep Lagrange Neural Network

DeLaN, a specialized variant of PINNs, integrates the Lagrangian equation into a deep learning architecture [10]. Compared to conventional system dynamics derived from kinematic assumptions and fundamental dynamic equations, DeLaN offers a more concise and efficient modeling process, requiring less detailed system information. To improve modeling accuracy and data sensitivity, this section presents an enhanced version of DeLaN augmented with a self-attention mechanism.

A. Attention Mechanism

The attention mechanism's basic principle is to identify the importance of each part of the input information by calculating a weight attention vector, then perform weighted summation of the input information according to these weights to obtain a weighted value. Such a weighted value helps emphasize

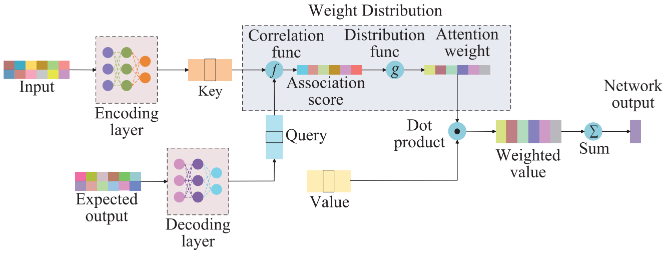


Fig. 1. Basic principle of attention mechanism.

critical information while filtering out irrelevant inputs, thereby enabling the neural network to perform tasks more efficiently. The basic principle of the attention mechanism is shown in Fig. 1.

The neural network encodes received data by converting input (text, images) into numerical vectors/tensors via input embedding and encoding layers (fully connected, convolutional, or recurrent). The resulting vector, termed the key (k), represents the input of the Weight Distribution module. Similarly, the expected output is estimated from the collected data, and then obtain the query vector (q) through a decoding layer, which reflects task information and guides the attention focus. q can be generated via model parameters, dynamically derived from input context (e.g., machine translation), extracted from input features through transformations, or predefined for specific tasks.

After obtaining k and q, the neural network calculates their "similarity" as the association score $f(q, k)$, reflecting the importance of q to k. This score is normalized to attention weight α by $g(\cdot)$. To get the attention output, the value vector v (providing specific input sequence info) is introduced, which is calculated by processing input through layers. v and k correspond one-to-one. The attention output y is then derived using $y = \beta(v, \alpha)$, where $\beta(\cdot)$ is often defined as $v^{\mathrm{T}}\alpha$.

Considering the characteristics of the parameters to be fitted and the complexity of the neural network, the correlation function $f(q, k) = q^{\mathrm{T}}k$ is used. The normalization function $g(\cdot)$ takes the Softmax function, such that, for the vector x, the i-th component of $g(x)$ is defined as $g_i(x) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$.

B. Self-Attention Enhanced Deep Lagrange Neural Network

Due to the inherent complexity and uncertainty of practical soft robots, acquiring prior motion knowledge is often challenging. As a result, selecting an appropriate query vector in the standard attention mechanism becomes difficult. To address this issue, this subsection integrates a self-attention mechanism into DeLaN to enhance dynamics learning.

The self-attention mechanism (see the left side of Fig. 2) originates from the basic attention mechanism but introduces essential differences. Instead of using inputs and expected outputs as the source of key and query, all the keys, queries and values in self-attention mechanism are from the same network input, processed through separate fully-connected layers. These keys and queries go through a Weight Distribution module and then dot product with values to generate the unknown dynamics functions. The input data includes pre-collected temporally continuous state-control pairs, and the weight distribution module is the same as that in Fig. 1. In the right side of Fig. 2, the output vectors of the self-attention mechanism are fed into the DeLaN prediction module. These vectors are reshaped into the estimated dynamics matrices, which are then used to compute the next-step states predictions via forward dynamics and discrete integration. The loss is evaluated as the discrepancy between predicted and real states, and the network is iteratively optimized using the Adamw optimizer with Autograd-based backpropagation. The detailed working principle of the DeLaN prediction module is presented in the remainder of this subsection.

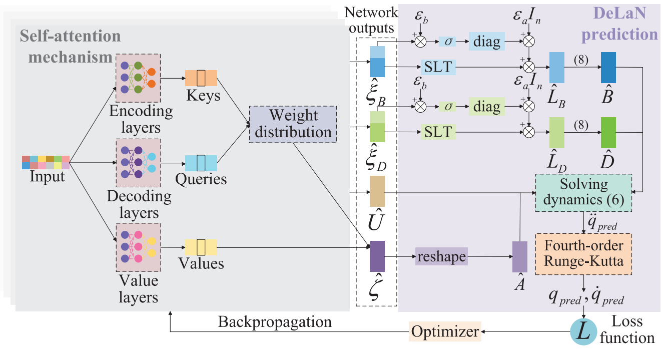


Fig. 2. DeLaN with self-attention mechanism.

According to [26], the dynamics of a continuum soft robot can be expressed as follows

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + K(q)q + D(q)\dot{q} + G(q) = A(q)u \tag{1}$$

where $q \in \mathbb{R}^n$ denotes configuration states; $B(q) \in \mathbb{R}^{n\times n}$ and $C(q, \dot{q}) \in \mathbb{R}^{n\times n}$ represent the inertial and Coriolis matrices, respectively; $K(q) \in \mathbb{R}^{n\times n}$ is the stiffness matrix, $D(q) \in \mathbb{R}^{n\times n}$ is the dissipation matrix, $G(q) \in \mathbb{R}^n$ denotes the gravity vector; and $A(q) \in \mathbb{R}^n$ is the input transform matrix. Given that $\dot{B} - 2C$ is skew-symmetric, the following equation holds

$$C(q, \dot{q}) = B(q)\dot{q} - \frac{1}{2}\left(\frac{\partial}{\partial q}(\dot{q}^{\mathrm{T}}B(q)\dot{q})\right)^{\mathrm{T}} \tag{2}$$

The derivatives of potential energy U with respect to q can be expressed as

$$\frac{\partial U}{\partial q} = K(q)q + G(q) \tag{3}$$

Integrating the Lagrange function, $L = \frac{1}{2}\dot{q}^{\mathrm{T}}B(q)\dot{q} - U(q)$, with (1), (2) and (3) yields

$$\ddot{q} = \left(\frac{\partial^2 L}{\partial \dot{q}^2}\right)^{-1}\left(A(q)u - \frac{\partial^2 L}{\partial q \partial \dot{q}}\dot{q} + \frac{\partial L}{\partial q} - D\dot{q}\right) \tag{4}$$

Design the second-order time derivative of the predicted configuration variable as follows:

$$\ddot{q}_{pred}(q, \dot{q}, t) = \left(\frac{\partial^2 \hat{L}}{\partial \dot{q}^2}\right)^{-1}\left(\hat{A}(q)u - \frac{\partial^2 \hat{L}}{\partial q \partial \dot{q}}\dot{q} + \frac{\partial \hat{L}}{\partial q} - \hat{D}\dot{q}\right) \tag{5}$$

where $\hat{L} = \frac{1}{2}\dot{q}^{\mathrm{T}}\hat{B}(q)\dot{q} - \hat{U}(q)$, $\hat{A}$ and $\hat{D}$ denote the estimated value of corresponding dynamics functions. Since $D(q), B(q) > 0$ are symmetric, according to the Cholesky decomposition theorem, they can be expressed as the product of the lower triangular matrix and its transpose

$$B(q) = L_B(q)L_B^{\mathrm{T}}(q), \quad D(q) = L_D(q)L_D^{\mathrm{T}}(q) \tag{6}$$

Accordingly, the estimated $B(q)$ and $D(q)$ are designed as

$$\begin{aligned}\hat{B}(q) &= \hat{L}_B(q)\hat{L}_B^{\mathrm{T}}(q) + \varepsilon I_n \\ \hat{D}(q) &= \hat{L}_D(q)\hat{L}_D^{\mathrm{T}}(q) + \varepsilon_a I_n\end{aligned} \tag{7}$$

in which $\hat{L}_B$, $\hat{L}_D$ are the estimated lower triangular matrices, $\varepsilon_a > 0$ is a small regularization constant, referred to as the diagonal matrix bias, introduced to prevent singularities in $\hat{B}(q)$, $\hat{D}(q)$.

The output of self-attention module includes four parts: $\hat{\xi}_B \in \mathbb{R}^p$, $\hat{\xi}_D \in \mathbb{R}^p$, $\hat{U} \in \mathbb{R}$ and $\hat{\zeta} \in \mathbb{R}^{n^2}$, where $p = \frac{n(n+1)}{2}$ is the number of learnable elements in $\hat{L}_B$, $\hat{L}_D$. Each of the four parts is learned by an independent self-attention module. $\hat{L}_B$ and $\hat{L}_D$ is calculated as follows:

$$\begin{aligned}\hat{L}_B &= \mathrm{SLT}(\hat{\xi}_{B2}) + \mathrm{diag}(\sigma(\hat{\xi}_{B1} + \varepsilon_b \mathbf{1}_n)) \\ \hat{L}_D &= \mathrm{SLT}(\hat{\xi}_{D2}) + \mathrm{diag}(\sigma(\hat{\xi}_{D1} + \varepsilon_b \mathbf{1}_n))\end{aligned} \tag{8}$$

in which $\hat{\xi}_{B1}, \hat{\xi}_{D1} \in \mathbb{R}^n$ are vectors of first n elements of $\hat{\xi}_B$ and $\hat{\xi}_D$. $\hat{\xi}_{B2}, \hat{\xi}_{D2} \in \mathbb{R}^{p-n}$ are vectors of the remaining elements of $\hat{\xi}_B$, $\hat{\xi}_D$, respectively. $\mathrm{SLT}(\cdot)$ is a reshaping function from a vector to strictly lower triangular matrix. $\sigma(\cdot)$ denotes the Softplus operator. $\varepsilon_b > 0$ is another regularization parameter, referred to as the diagonal matrix shift. The estimated potential energy $\hat{U}$ is directly output, while $\hat{\zeta}$ has to be reshaped into the estimated input transform matrix $\hat{A} \in \mathbb{R}^{n\times n}$ row by row.

The loss is defined as the Euclidean distance between the predicted and actual states at the next time step, *s.t.*

$$Loss = \|x_{next} - x_{pred}\|^2 \tag{9}$$

where $x_{next} = [q_{next}; \dot{q}_{next}]$ and $x_{pred} = [q_{pred}; \dot{q}_{pred}]$ are the true and predicted vectors composed of the robot configuration variables and their time derivatives. The proposed dynamics learning neural network aims at adjusting all learnable parameters $\boldsymbol{\varphi}$ to minimize the fitting error

$$\boldsymbol{\varphi} = \underset{\boldsymbol{\varphi}}{\mathrm{argmin}}\, Loss \tag{10}$$

The optimization problem is solved through built-in Autograd and backpropagation optimizers. The whole self-attention enhanced DeLaN is mainly written in JAX module of Python.

III. Adaptive Fractional-Order Shape Control for Soft Robots

This section presents the controller design (shown in Fig. 3). First, a NDOB is proposed to estimate and compensate for external disturbances and uncertainties of the learned model. Subsequently, an adaptive fractional-order sliding mode controller is proposed, following which the closed-loop stability is proved.

A. Preliminaries

Before outlining the formal design process, relevant definitions, properties and lemmas are first introduced in this subsection.

Definition 1 [27]: The α-order Caputo differential is defined as follows

$${}_C D_t^{\alpha} f(t) = \begin{cases} \frac{d^m f(t)}{dt^m}, & \alpha = m \\ \frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{D^m f(\tau)}{(t-\tau)^{\alpha-m+1}} d\tau, & m-1<\alpha<m \end{cases} \tag{11}$$

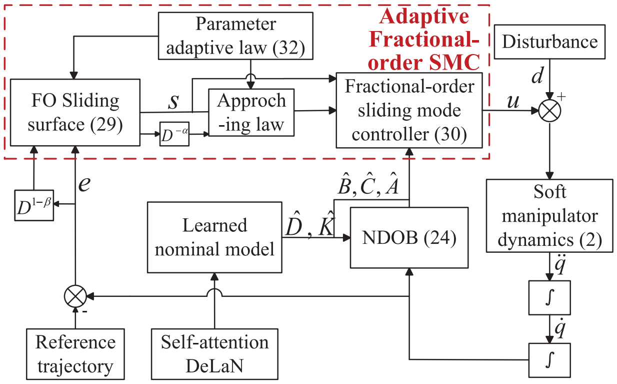


Fig. 3. The diagram of the proposed controller.

where $\alpha \in \mathbb{R}^+$, $m \in \mathbb{N}^+$, $\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt$ is the Gamma function. Hereafter, the fractional operator ${}_C D_t^{\alpha}$ will be abbreviated as D_t^{α}. When there is no ambiguity, D^{α} is used instead of D_t^{α}. In addition, for positive numbers α and ϵ that are not integers, the definition of α-order ϵ-Caputo calculus is as follows:

$${}_{\epsilon} D_t^{\alpha} f(t) = \frac{1}{\Gamma(m-\alpha)} \int_{\max(t-\epsilon,0)}^{t} \frac{D^m f(\tau)}{(t-\tau)^{\alpha-m+1}} d\tau \tag{12}$$

in which m is an integer that satisfies $m-1 < \alpha < m$.

Definition 2 [28]: The α-order integral and ϵ-integral are defined as follows

$$\begin{aligned} D_t^{-\alpha} f(t) &= \frac{1}{\Gamma(\alpha)} \int_0^t \frac{f(\tau)}{(t-\tau)^{1-\alpha}} d\tau \\ {}_{\epsilon} D_t^{-\alpha} f(t) &= \frac{1}{\Gamma(\alpha)} \int_{\max(t-\epsilon,0)}^{t} \frac{f(\tau)}{(t-\tau)^{1-\alpha}} d\tau \end{aligned} \tag{13}$$

where $0 < \alpha < 1$ and $\epsilon > 0$.

Property 1: Caputo fractional calculus has a semi-group property, *i.e.*, for positive real numbers α, β satisfying $\alpha+\beta \le 1$ and any $f(t)$ that makes the following equation meaningful, equation $D^{\beta}(D^{\alpha} f(t)) = D^{\alpha+\beta} f(t)$ holds. If α or β is negative, appropriate initial conditions of $f(t)$ are required.

Lemma 1 [29]: For a continuous-time autonomous system $\dot{x}(t) = f(x(t))$ with equilibrium point $x = 0$ and a positive definite and continuously differentiable Lyapunov function candidate $V(x)$, if $\dot{V}(x) + aV(x) + bV^{\sigma}(x) \le 0$, then the system is finite-time stable and converges within time

$$T \le \frac{1}{a(1-\sigma)} \ln\left(\frac{aV^{1-\sigma}(x_0) + b}{b}\right) \tag{14}$$

where $a, b > 0, 0 < \sigma < 1$. Specifically, if $a = 0$, then the system is also finite-time stable with convergence time $T \le \frac{V^{1-\sigma}(x_0)}{b(1-\sigma)}$.

Lemma 2 [27]: Let $R(c)$ denote the real part of c. If $R(\alpha) > 0$, $D_{a+}^{-\alpha}$ and $D_{b-}^{-\alpha}$ are bounded in $L_p(a, b)$, then for any $1 \le p \le +\infty$ and $f(x)$, we have

$$\|D_{a+}^{-\alpha} f(x)\|_p \le K\|f(x)\|_p, \|D_{b-}^{-\alpha} f(x)\|_p \le K\|f(x)\|_p \tag{15}$$

in which $K = \frac{(b-a)^{R(\alpha)}}{R(\alpha)|\Gamma(\alpha)|}$, $\Gamma(\cdot)$ is the Gamma function.

Lemma 3 [30]: Considering the saturation function

$$\mathrm{sat}_\varepsilon(x) = \begin{cases} 1, & x > \varepsilon \\ \frac{1}{\varepsilon}, & -\varepsilon \le x \le \varepsilon \\ -1, & x < -\varepsilon \end{cases} \tag{16}$$

in which $\varepsilon > 0$. For any continuously differentiable $x(t)$, when $|x| \ge \varepsilon$, $xD_t^\gamma \mathrm{sat}_\varepsilon(x) \ge 0$ holds for any $0 < \gamma < 1$.

Lemma 4: For matrices $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^{n\times 1}$, $c \in \mathbb{R}^{n\times 1}$, we have

$$\begin{aligned} \lambda_{\min}(A)\|b\|^2 \le \|b^{\mathrm{T}}Ab\| \le \lambda_{\max}(A)\|b\|^2 \\ \|c^{\mathrm{T}}Ab\| \le \lambda_{\max}(A)\|b\|\|c\| \end{aligned} \tag{17}$$

where $A > 0$, $\lambda_{\min}(\cdot)$, $\lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalue, $\|\cdot\|$ means the L_2-norm of vector.

B. Design of NDOB

Model complexity and imperfect generalization of the neural networks are prone to introducing uncertainties, further consider the external disturbance d, the practical system dynamics is written as

$$\begin{aligned} &(\bar{B} + \delta B)\ddot{q} + (\bar{C} + \delta C)\dot{q} + (\bar{K} + \delta K)q + (\bar{D} + \delta D)\dot{q} \\ &\quad + (\bar{G} + \delta G) = (\bar{A} + \delta A)(u + d) \end{aligned} \tag{18}$$

where $\bar{\cdot}$ represents the known nominal system matrices provide by the self-attention enhanced DeLaN, $\delta\cdot$ stands for the model uncertainties distributed in system matrices. Let $\bar{\mathcal{H}}$ collect the nominal parts and $\delta\mathcal{H}$ the uncertain terms in the left side of (18), one has $\bar{\mathcal{H}} + \delta\mathcal{H} - (\bar{A} + \delta A)d = (\bar{A} + \delta A)u$. Mapping the lump uncertainties to the input yields

$$\delta\mathcal{F} = \frac{\bar{\mathcal{H}}}{\bar{A}} + d - \frac{\bar{\mathcal{H}} + \delta\mathcal{H}}{\bar{A} + \delta A} = \frac{\bar{\mathcal{H}}\delta A - \bar{A}\delta\mathcal{H}}{(\bar{A} + \delta A)\bar{A}} + d \tag{19}$$

The goal is to estimate $\delta\mathcal{F}$, denoted $\delta\hat{\mathcal{F}}$, and compensate it at the input end, *i.e.,* $u = u^* - \delta\hat{\mathcal{F}}$. From (1), the system dynamics with equivalent input disturbance is expressed as

$$\dot{x} = \bar{f}(x) + \bar{g}(x)(u + \delta\mathcal{F}) \tag{20}$$

where $x = [q; \dot{q}]$, $\bar{f}(x) = [\dot{q}; -\bar{B}^{-1}(-\bar{C}\dot{q} - \bar{K}q - \bar{D}\dot{q} - \bar{G})]$, $\bar{g}(x) = [\mathbf{0}; \bar{B}^{-1}\bar{A}]$. The NDOB is designed as

$$\begin{aligned} &\delta\hat{\mathcal{F}} = z + p(x) \\ &\dot{z} = l(x)(-\bar{f}(x) - \bar{g}(x)u - \bar{g}(x)p(x)) - l(x)\bar{g}(x)z \end{aligned} \tag{21}$$

where $p(x)$ and $l(x)$ satisfy $\frac{\partial p(x)}{\partial x} = l(x)$ and $l(x) = [\mathbf{0}, \Lambda\bar{A}^{-1}\bar{B}]$, $\Lambda > 0$ is a positive definite constant matrix, determining the convergence speed of NDOB.

Assumption 1: Given that the constituent materials of soft robots are flexible, the shape of soft robots should change in a continuous manner. Additionally, as weights chosen in self-attention enhanced DeLaN are finite because of the weight decay term added in the loss function, it is reasonable to assume that both the model uncertainties and their time derivatives are bounded. Moreover, since the external energy applied to soft robots is finite, according to assumptions mentioned in [31], the external disturbance and its disturbance applied on soft robots are also finite. Consequently, we assume that $\delta\mathcal{F}$, $\delta\dot{\mathcal{F}}$ are bounded by $\|\delta\mathcal{F}\| < \mathcal{F}_{max}$ and $\|\delta\dot{\mathcal{F}}\| < \dot{\mathcal{F}}_{max}$.

Theorem 1: Considering the state-space system dynamics (20) with the proposed disturbance observer (21), the estimation error $\delta\tilde{\mathcal{F}} = \delta\mathcal{F} - \delta\hat{\mathcal{F}}$ is finite time stable.

Proof: Consider the following Lyapunov candidate

$$V_f = \frac{1}{2}\delta\tilde{\mathcal{F}}^{\mathrm{T}}\Lambda^{-1}\delta\tilde{\mathcal{F}} \tag{22}$$

its time derivative can be calculated by

$$\begin{aligned} \dot{V}_f &= \delta\tilde{\mathcal{F}}^{\mathrm{T}}\Lambda^{-1}\delta\dot{\tilde{\mathcal{F}}} = \delta\tilde{\mathcal{F}}^{\mathrm{T}}\Lambda^{-1}(\delta\dot{\mathcal{F}} - \delta\dot{\hat{\mathcal{F}}}) \\ &= \delta\tilde{\mathcal{F}}^{\mathrm{T}}\Lambda^{-1}(\delta\dot{\mathcal{F}} - l(x)(-\bar{f}(x) - \bar{g}(x)u - \bar{g}(x)p(x)) \\ &\quad + l(x)\bar{g}(x)z - l(x)\dot{x}) \\ &= \delta\tilde{\mathcal{F}}^{\mathrm{T}}\Lambda^{-1}(\delta\dot{\mathcal{F}} - l(x)(-\bar{f}(x) - \bar{g}(x)u - \bar{g}(x)\delta\hat{\mathcal{F}} + \dot{x})) \\ &= \delta\tilde{\mathcal{F}}^{\mathrm{T}}\Lambda^{-1}\delta\dot{\mathcal{F}} - \delta\tilde{\mathcal{F}}^{\mathrm{T}}\delta\tilde{\mathcal{F}} \end{aligned} \tag{23}$$

According to Lemma 4, $\Lambda^{-1}\delta\dot{\mathcal{F}} \le \lambda_{\min}^{-1}(\Lambda)\dot{\mathcal{F}}_{max}$, where $\lambda_{\min}(\Lambda)$ denotes the minimum eigenvalue of Λ. When $\|\delta\tilde{\mathcal{F}}\| \ge 2\lambda_{\min}^{-1}(\Lambda)\dot{\mathcal{F}}_{max}$, $\dot{V}_f \le -\dot{\mathcal{F}}_{max}\lambda_{\min}^{-1}(\Lambda)\|\delta\tilde{\mathcal{F}}\| \le -\sqrt{2}\dot{\mathcal{F}}_{max}\lambda_{\min}^{-\frac{1}{2}}(\Lambda)V_f^{\frac{1}{2}}$. From Lemma 1, estimated errors converge to $\|\delta\tilde{\mathcal{F}}\| \le 2\dot{\mathcal{F}}_{max}\lambda_{\min}^{-1}(\Lambda)$ in finite time. Choosing a larger $\lambda_{\min}(\Lambda)$ can accelerate the convergence rate and reduce the ultimate convergence neighborhood.

This completes the proof of Theorem 1.■

Remark 1: To tackle fast-varying situations, adding an additional compensation term to (21) yields

$$\begin{aligned} \dot{z} = l(x)(-\hat{f}(x) - \hat{g}(x)u - \hat{g}(x)p(x)) \\ - l(x)\hat{g}(x)z + \alpha\delta\dot{\hat{\mathcal{F}}}^* \end{aligned} \tag{24}$$

where $*$ denotes a linear combination of previous states

$$x^* = \sum_{i=1}^{r} a_i x^{[-i]} \tag{25}$$

in which r is the number of prediction steps, $x^{[-k]}$ represents the state quantity before k steps, a_i represents the weight of each history state and $0 < \alpha < 1$ is the attenuation coefficient. Increasing α leads to better tracking ability for fast-varying uncertainties. However, setting α too large is prone to oscillation or even divergence.

C. Adaptive Fractional-Order Controller Design

A novel fractional-order sliding surface is designed as

$$s = \dot{e} + k_1 e + k_{2\epsilon}D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e) + \mathrm{sig}^{1+\nu}(e)) \tag{26}$$

in which $e = q_{ref} - q$ represents the tracking error $k_1, k_2 > 0, 0 < \beta, \nu < 1$ are the surface parameters, $\mathrm{sig}^\nu(x) = |x|^\nu \mathrm{sgn}(x)$ with $\mathrm{sgn}(x)$ the sign function. The control input is designed as follows

$$\begin{aligned} u &= (A(q))^{-1}\left(\frac{\partial^2 L}{\partial\dot{q}^2}\ddot{q} + \frac{\partial^2 L}{\partial q\partial\dot{q}}\dot{q} + D(q)\dot{q} - \frac{\partial L}{\partial q}\right) - \delta\hat{\mathcal{F}} \\ &= (A(q))^{-1}(\frac{\partial^2 L}{\partial\dot{q}^2}(\ddot{q}_{ref} + k_1\dot{e} + k_{2\epsilon}D_t^{1-\beta}(\mathrm{sig}^{1-\nu}(e) \\ &\quad + \mathrm{sig}^{1+\nu}(e)) - (f_1(s) + D^{-\alpha}(f_2(s)))) + \frac{\partial^2 L}{\partial q\partial\dot{q}}\dot{q} \\ &\quad + D(q)\dot{q} - \frac{\partial L}{\partial q}) - \delta\hat{\mathcal{F}} \end{aligned} \tag{27}$$

where

$$\begin{aligned} f_1(s) &= -k_3\mathrm{sat}_\varepsilon(s) - k_4 s - k_5|s|^\rho D_t^\gamma \mathrm{sat}_\varepsilon(s) \\ &\quad - k_6 s\log(s+\sqrt{s^2+1})D_t^\gamma \mathrm{sat}_\varepsilon(s) \\ f_2(s) &= -k_7 s - k_8\mathrm{sat}_\varepsilon(s) \end{aligned} \tag{28}$$

represent the two parts of the sliding surface's reaching law, $0<\alpha,\gamma,\rho<1$, $\varepsilon>0$, $\mathrm{sat}_\varepsilon(s)$ is defined in Lemma 3. L, $D(q)$, and $A(q)$ are learned by the attention enhanced DeLaN. $k_i>0$, $i=3,4,\cdots,8$, k_3 and k_6 are adaptive satisfying

$$\begin{cases} \dot{k}_3 &= \begin{cases} 0, & \|s\|\le\varepsilon_1 \ \cap\ k_3=k_{3min} \\ 0, & \|s\|\ge\varepsilon_1 \ \cap\ k_3=k_{3max} \\ w_1\|s\|\mathrm{sgn}(\|s\|-\varepsilon_1), & \text{Otherwise} \end{cases} \\ \dot{k}_6 &= \begin{cases} 0, & \|s\|\le\varepsilon_2 \ \cap\ k_6=k_{6min} \\ 0, & \|s\|\ge\varepsilon_2 \ \cap\ k_6=k_{6max} \\ w_2 e^{-\|s\|}\mathrm{sgn}(\|s\|-\varepsilon_2), & \text{Otherwise} \end{cases} \end{cases} \tag{29}$$

in which $w_1, w_2, \varepsilon_1, \varepsilon_2, k_{3min}, k_{3max}, k_{6min}, k_{6max} > 0$. From (29), k_3 and k_6 are within $[k_{3min}, k_{3max}]$ and $[k_{6min}, k_{6max}]$, respectively.

Remark 2: The proposed controller strikes a balance between rapid convergence and robust performance by introducing a novel sliding surface, reaching law, and parameter-adaptation law. The sliding surface, incorporating two distinct nonlinear terms, can flexibly address errors of varying magnitudes. The reaching law fuses fractional-order and integer-order dynamics, fractional calculus offers a wider frequency-band response and adaptive dynamic tuning, while integer-order control ensures stable asymptotic behavior. Together, they greatly suppress oscillations without compromising overall stability. In addition, the parameters adaptation is stage dependent. When errors are large, the controller emphasizes speed, accelerating convergence. As the system approaches its target, it shifts focus to robustness and disturbance rejection, maintaining stability in the steady state.

Theorem 2: Considering system (20) subject to lumped uncertainties $\delta\mathcal{F}$, under controller (27) and disturbance observer (21), the sliding surface (26) is finite time convergence.

Proof: Substituting (27) and (26) into (20) yields

$$\dot{s} = f_1(s) + D^{-\alpha}f_2(s) - \bar{g}(x)\delta\tilde{\mathcal{F}} \tag{30}$$

Step 1: Considering the Lyapunov candidate $V_1 = \frac{1}{2}s^{\mathrm{T}}s + \frac{1}{2w_1}k_3^{+2} + \frac{1}{2w_2}k_6^{+2}$, where $k_3^+ = k_3 - k_{3max} \le 0$, $k_6^+ = k_6 - k_{6max} \le 0$. Since $\bar{g}(x) = \bar{B}^{-1}\bar{A}$ includes only the nominal input transform matrix and inertial matrix, we assume that $\bar{g}(x)$ is bounded with its maximum eigenvalue $\max(\mathrm{abs}(\lambda_{\hat{g}(x)})) < N_g$. According to (30), we have

$$\begin{aligned} \dot{V}_1 &= s^{\mathrm{T}}\dot{s} + \frac{1}{w_1}k_3^+\dot{k}_3 + \frac{1}{w_1}k_6^+\dot{k}_6 \\ &= s^{\mathrm{T}}f_1(s) + s^{\mathrm{T}}D^{-\alpha}f_2(s) - \delta\tilde{\mathcal{F}}^{\mathrm{T}}\bar{g}(x)s + k_3^+\dot{k}_3 + k_6^+\dot{k}_6 \end{aligned} \tag{31}$$

From Lemma 2, there exists $K_1 > 0 s.t. \|D^{-\alpha}f_2(s)\| \le K_1\|f_2(s)\|$. From Theorem 1, there exists $\mathcal{F}_0 > 0 s.t. \|\delta\tilde{\mathcal{F}}\| < \mathcal{F}_0$. Using Lemma 4, $\delta\tilde{\mathcal{F}}^{\mathrm{T}}\bar{g}(x)s \le N_g\|\delta\tilde{\mathcal{F}}\|\|s\| \le N_g\mathcal{F}_0\|s\|$. Since $k_3^+, k_6^+ \le 0$, $\dot{k}_3, \dot{k}_6 \ge 0$, when $\|s\| \ge \max(\varepsilon_1, \varepsilon_2)$

$$\dot{V}_1 \le s^{\mathrm{T}}f_1(s) + K_1\|s\|\|f_2(s)\| + N_g\mathcal{F}_0\|s\| \tag{32}$$

holds. According to [30], $s_i D_t^\gamma \mathrm{sat}_\epsilon(s_i) \ge 0$, and $\log(s_i + \sqrt{s_i^2+1})$ shares the same sign as s_i, it follows that $s^{\mathrm{T}}(|s|^\rho D_t^\gamma \mathrm{sat}_\varepsilon(s)) > 0$ and $s^{\mathrm{T}}(s\log(s+\sqrt{s^2+1})D_t^\gamma \mathrm{sat}_\varepsilon(s)) > 0$. Therefore, we have

$$\begin{aligned} \dot{V}_1 &\le -s^{\mathrm{T}}(k_3\mathrm{sat}_\varepsilon(s) + k_4 s) \\ &\quad + K_1 s^{\mathrm{T}}(k_7\mathrm{sat}_\varepsilon(s) + k_8 s) + N_g\mathcal{F}_0\|s\| \\ &= -(k_3 - K_1k_7)s^{\mathrm{T}}\mathrm{sat}_\varepsilon(s) - (k_4 - K_1k_8)\|s\|^2 \\ &\quad + N_g\mathcal{F}_0\|s\| \end{aligned} \tag{33}$$

When $\|s\| > \varepsilon$, $s^{\mathrm{T}}\mathrm{sat}_\varepsilon(s) = \sum_{i=1}^n |s_i| \ge \|s\|$. As $k_3^{+2} \le (k_{3max} - k_{3min})^2$, $k_6^{+2} \le (k_{6max} - k_{6min})^2$ are both bounded, let $K_2 = (k_{3max} - k_{3min})^2 + (k_{6max} - k_{6min})^2$, we have $s^{\mathrm{T}}s \le V_1 \le s^{\mathrm{T}}s + K_2$. Then

$$\begin{aligned} \dot{V}_1 &\le -(k_3 - K_1k_7 - N_g\mathcal{F}_0)\|s\| - (k_4 - K_1k_8)\|s\|^2 \\ &\le -L_1(V_1 - K_2)^{\frac{1}{2}} - L_2(V_1 - K_2) \end{aligned} \tag{34}$$

where $L_1 = \sqrt{2}(k_3 - K_1k_7 - N_g\mathcal{F}_0)$, $L_2 = k_4 - K_1k_8$. Choosing $k_3 > K_1k_7 + N_g\mathcal{F}_0$ and $k_4 > K_1k_8$, then $L_1, L_2 > 0$. In this condition, if $V_1 \ge 2K_2$, then

$$\dot{V}_1 \le -\frac{L_1}{2}V_1^{\frac{1}{2}} - \frac{L_2}{2}V_1 \tag{35}$$

From Lemma 1, the proposed controller guarantees the sliding surface converges to $\|s\| < 2\sqrt{K_2}$ in finite-time.

Step 2: For the closed-loop system, consider $V_2 = V_f + V_1$, where V_f is defined in (22). Taking (30) yields

$$\begin{aligned} \dot{V}_2 &\le \dot{V}_f + s^{\mathrm{T}}(f_1(s) + D^{-\alpha}f_2(s)) - \delta\tilde{\mathcal{F}}^{\mathrm{T}}\bar{g}(x)s \\ &= \delta\tilde{\mathcal{F}}^{\mathrm{T}}\Lambda^{-1}\delta\dot{\tilde{\mathcal{F}}} - \delta\tilde{\mathcal{F}}^{\mathrm{T}}\delta\tilde{\mathcal{F}} - \delta\tilde{\mathcal{F}}^{\mathrm{T}}\bar{g}(x)s \\ &\quad + s^{\mathrm{T}}(f_1(s) + D^{-\alpha}f_2(s)) \\ &\le \|\delta\tilde{\mathcal{F}}\|\dot{\mathcal{F}}_{max}\lambda_{\min}^{-1}(\Lambda) - \|\delta\tilde{\mathcal{F}}\|^2 + N_g\|\delta\tilde{\mathcal{F}}\|\|s\| \\ &\quad - s^{\mathrm{T}}(k_3\mathrm{sat}_\varepsilon(s) + k_4 s) + K_1 s^{\mathrm{T}}(k_7\mathrm{sat}_\varepsilon(s) + k_8 s) \end{aligned} \tag{36}$$

Let $k_3 - K_1k_7 > 0$ and $k_4 - K_1k_8 > 0$, one has

$$\begin{aligned} \dot{V}_2 &\le -\left(\frac{\|\delta\tilde{\mathcal{F}}\|}{2} - N_g\|s\|\right)^2 + \dot{\mathcal{F}}_{max}\lambda_{\min}^{-1}(\Lambda)\|\delta\tilde{\mathcal{F}}\| - \frac{3}{4}\|\delta\tilde{\mathcal{F}}\|^2 \\ &\quad - (k_3 - K_1k_7)s^{\mathrm{T}}\mathrm{sat}_\varepsilon(s) - (k_4 - K_1k_8)\|s\|^2 \\ &\le \dot{\mathcal{F}}_{max}\lambda_{\min}^{-1}(\Lambda)\|\delta\tilde{\mathcal{F}}\| - \frac{3}{4}\|\delta\tilde{\mathcal{F}}\|^2 - (k_4 - K_1k_8)\|s\|^2 \\ &\le \frac{\dot{\mathcal{F}}_{max}}{\lambda_{\min}(\Lambda)}\sqrt{\lambda_{\max}(\Lambda)V_f} - \frac{3}{4}\lambda_{\min}(\Lambda)V_f - (k_4 - K_1k_8)s^{\mathrm{T}}s \\ &\le K_3V_2^{\frac{1}{2}} - K_4V_2 \end{aligned} \tag{37}$$

As $\sqrt{V_f} \le V_2^{\frac{1}{2}}$, $V_f + s^{\mathrm{T}}s \ge V_2 - K_2$, it yields

$$\dot{V}_2 \le K_3V_2^{\frac{1}{2}} - K_4V_2 + K_3K_2 \tag{38}$$

where $K_3 = \frac{\dot{\mathcal{F}}_{max}}{\lambda_{\min}(\Lambda)}\sqrt{\lambda_{\max}(\Lambda)}$, $K_4 = \min\left(\frac{3}{4}\lambda_{\min}(\Lambda), 2(k_4 - K_1k_8)\right)$. When $V_2 \ge \left(\frac{K_3 + \sqrt{K_3^2 + 2K_4^2K_2}}{K_4}\right)^2$,

$\dot{V}_2 \leq -\frac{K_3+\sqrt{K_3^2+2K_4^2K_2}}{2}V_2^{\frac{1}{2}}$. From Lemma 1, the system will converge to $\|s\| < \max\left(\sqrt{2}\frac{K_3+\sqrt{K_3^2+2K_4^2K_2}}{K_4}, \varepsilon_1, \varepsilon_2\right)$ in finite time.

This completes the proof of Theorem 2.■

Remark 3: Proper parameters selection can effectively reduce the convergence region. An optimal choice is $\Lambda = \alpha_\Lambda I_n > 0$, which yields $\lambda_{\max}(\Lambda) = \lambda_{\min}(\Lambda) = \alpha_\Lambda > 0$. Moreover, increasing k_3 and k_4 can accelerate convergence and improve tracking accuracy, this may also induce greater oscillations. As $\alpha_\Lambda, k_4 \to +\infty$, it follows that $K_3 \to 0$, $K_3 \to +\infty$, and consequently $\frac{K_4}{2} \to +\infty$. In this case, the radius of convergence region $\sqrt{2}\frac{K_3+\sqrt{K_3^2+2K_4^2K_2}}{K_4} \to 2\sqrt{K_2}$.

Theorem 3: When the sliding surface s is in the neighborhood of the origin, *i.e.* $|s_i| < \epsilon_\iota$, then the tracking error e_i and its derivative $\dot{e}_i$ will be stable in finite time.

Proof: For each element s_i, e_i, let the Lyapunov candidate $V_{3i} = \frac{1}{2}e_i^2 \geq 0$. After the convergence of s_i, we have

$$\dot{V}_{3i} = -k_1e_i^2 - k_2e_{i\epsilon}D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i)) + e_i\iota \tag{39}$$

where $|\iota| < \epsilon_\iota$. First, assuming that ${}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i)) \neq 0$. If there exists an instant $t = t_z$ such that ${}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i)) = 0$, then we can find a punctured neighborhood of t_z where $\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i)) \neq 0$ holds. Otherwise, there exists a time interval containing t_z in which ${}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i)) = 0$, it follows that $e_i = \dot{e}_i = 0$, leading to stability of the system. According to the definition of α-order ϵ-Caputo calculus, we have

$$\begin{aligned}
&|{}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i))| \\
&= \frac{1}{\Gamma(\beta)}|\int_{\max(t-\epsilon,0)}^{t} \frac{\mathrm{sig}^{1-\nu}(e_i(\tau)) + \mathrm{sig}^{1+\nu}(e_i(\tau))}{(\epsilon-\tau)^{1-\beta}}d\tau| \\
&\leq \frac{1}{\Gamma(\beta)}|\left(\int_{\max(t-\epsilon,0)}^{t} \frac{1}{(\epsilon-\tau)^{1-\beta}}d\tau\right)(e_{i,m}^{1-\nu} + e_{i,m}^{1+\nu})| \\
&= \frac{\epsilon^\beta}{\beta\Gamma(\beta)}(e_{i,m}^{1-\nu} + e_{i,m}^{1+\nu})
\end{aligned} \tag{40}$$

where $e_{i,m} > 0$ denotes the supremum of $|e_i|$ in the time period of $(t-\epsilon, t]$. Then for a pre-defined boundary value δ, the proof is divided into two cases.

Case 1: $|\frac{\epsilon_\iota}{{}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i)+\mathrm{sig}^{1+\nu}(e_i))}| < \delta$. Let $k_2 > \delta$ and $\kappa_i = k_2 - \frac{\iota}{{}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i)+\mathrm{sig}^{1+\nu}(e_i))} > 0$, then (39) can be transformed into

$$\begin{aligned}
\dot{V}_{3i} &= -k_1e_i^2 - \kappa_ie_{i\epsilon}D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i)) \\
&\leq -k_1e_i^2 - \kappa\zeta|e_i| = -2k_1V_{3i} - \sqrt{2}\kappa\zeta V_{3i}^{0.5}
\end{aligned} \tag{41}$$

where constant ζ satisfies $0 < \zeta < |{}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i))|$. If $e_i \neq 0$, according to Definition 1, $e_{i\epsilon}D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i)+\mathrm{sig}^{1+\nu}(e_i)) > 0$ as long as ϵ is small enough. Then such ζ exists. For an instant when $e_i = 0$, we can find the neighborhood around this time where $e_i \neq 0$. Otherwise, $e_i = 0, \dot{e}_i = 0$ implies system's stability. From Lemma 1, e_i converges in finite time.

Case 2: $|\frac{\epsilon_\iota}{{}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i)+\mathrm{sig}^{1+\nu}(e_i))}| \geq \delta$. We have

$$|{}_\epsilon D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i))| \leq |\frac{\iota}{\delta}| < \frac{\epsilon_\iota}{\delta} \tag{42}$$

TABLE I

SYSTEM PARAMETERS

System Parameters	Value
Initial length of every segment L/m	1
Valid radius r/m	0.05
Chamber radius r_c/m	0.01
Linear density ρ_l/(kg/m)	0.3
Gravity g/(kg·m/s^2)	-9.81 at z axis
Stiffness matrix K	eye(6)
Damping matrix D	0.1×eye(6)

From (39), it is clear that

$$\dot{V}_{3i} \leq -k_1e_i^2 + e_i\left(\frac{k_2\epsilon_\iota}{\delta} + \epsilon_\iota\right) \tag{43}$$

If $e_i \geq 2\frac{k_2\epsilon_\iota+\epsilon_\iota\delta}{k_1\delta}$, then $\dot{V}_{3i} \leq -\frac{k_2\epsilon_\iota+\epsilon_\iota\delta}{\delta}e_i$, indicating finite-time stability of the system. The tracking error e_i can converge to the region $(-\Upsilon_i, \Upsilon_i)$ in finite time, where

$$\Upsilon_i = 2\frac{k_2\epsilon_\iota + \epsilon_\iota\delta}{k_1\delta} \tag{44}$$

From (40), $\dot{e}_i$ satisfies

$$\begin{aligned}
|\dot{e}_i| &= |-k_1e_i - k_{2\epsilon}D_t^{-\beta}(\mathrm{sig}^{1-\nu}(e_i) + \mathrm{sig}^{1+\nu}(e_i)) + \iota| \\
&\leq k_1\Upsilon_i + k_2\frac{\epsilon^\beta}{\beta\Gamma(\beta)}(\Upsilon_i^{1-\nu} + \Upsilon_i^{1+\nu}) + \epsilon_\iota
\end{aligned} \tag{45}$$

indicating that e_i and $\dot{e}_i$ is also finite time stable.

This completes the proof of Theorem 3.■

Remark 4: Since the right hand side of (44) approaches zero as $k_1 \to \infty$, increasing k_1 helps reduce tracking error. However, an excessively large k_1 amplifies $\dot{e}_1$, leading to oscillations near the equilibrium. Both e_i and $\dot{e}_i$ are both positively correlated with k_2, though a larger k_2 shortens convergence time in Case 1. Thus, k_1 and δ should be chosen moderately, with k_2 slightly exceeding the preset δ. Moreover, Theorem 2 confirms that ϵ_ι can become arbitrarily small with appropriate controller and observer parameters.

Remark 5: The fractional order $0 < \beta < 1$ in the control scheme significantly influences control robustness and accuracy. If β is chosen too close to 1, the controller behaves similarly to an integer-order one, becoming more sensitive to high-frequency noise and disturbances. Conversely, setting β too close 0 may introduce excessive damping, leading to a slower dynamic response to reference variation, as the strong emphasis on historical states acts like a low-pass filter. In practice, choosing β within $0.4 \leq \beta \leq 0.7$ is generally suitable for balancing responsiveness and robustness.

IV. SIMULATION RESULTS

In this section, comparative simulations on a two-segments (with 6 DoFs) fully-actuated extensible continuum soft arm are conducted, in which the start point is fixed and end-effector points downwards. System parameters of the soft robots are listed in Table I. To avoid singularity, the Δ-parametrization approach [32] is adopted. The Oustaloup integer-order approximation [33] is utilized when calculate the fractional order calculus, where $N = 6$, $\mu = 10$, $\eta = 1$, $\omega_d = 1$.

TABLE II
HYPERPARAMETERS LIST OF DELAN

Hyperparameters	**Value**
Number of robot actuators n_{act}	6
Network width of Lagrangian w_l	42
Network depth of Lagrangian d_l	3
Network width of dissipative matrix w_d	5
Network depth of dissipative matrix d_d	2
Network width of input transform matrix w_i	42
Network depth of input transform matrix d_i	2
Activation function of Lagrangian	softplus
Activation function of dissipative matrix	tanh
Activation function of input transform matrix	sigmoid
Initial learning rate η	1.56×10^{-3}
Weight decay W_o	1.2×10^{-5}

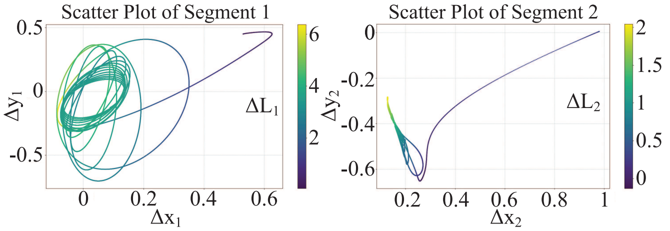
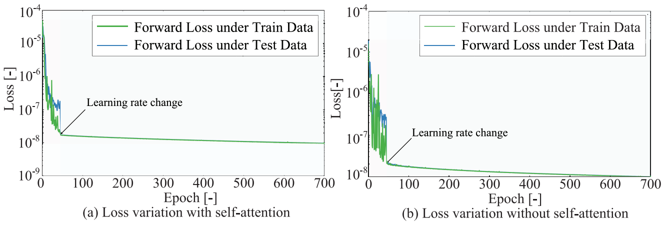


Fig. 4. Scattered thermogram visualization of 60,000 data points on the Δx-Δy plane. Δx_1, Δy_1, ΔL_1, Δx_2, Δy_2, ΔL_2 are the configuration states according to the Δ-parametrization in [32].



(a) Loss variation with self-attention

(b) Loss variation without self-attention

Fig. 5. Loss variation curves. A learning rate change happens at the 43*th* epoch, which reduces to 1.56×10^{-4} from initial value listed in Table II. The light red and light blue background represents epochs before and after learning rate change, respectively.

A. Performance of the Self-Attention Enhanced DeLaN in Dynamics Learning for Soft Robots

The hyperparameters of DeLaN are listed in Table II, whereas the learning rate of DeLaN is adaptive, gradually reduces during the learning process.

The designed self-attention enhanced DeLaN designed uses 60,000 sets of temporally continuous data as samples, where each set of data includes input u and six-dimensional configuration variables with their time derivatives q, $\dot{q}$, $\ddot{q}$. In this neural network, 70% of the random data is used as training data, and the remaining 30% are used as test data. In Fig. 4, the collected data of configuration states is visualized. The sampling period is set as $\Delta t = 10^{-3}$ s.

First, set the number of training epochs as 700. Fig. 5(a) and Fig. 5(b) show the loss function curves over epochs for DeLaN with and without attention. Obviously, compared to the traditional DeLaN, the loss function converges faster and shows less fluctuation under the self-attention enhanced DeLaN, especially in the training set, demonstrating the effectiveness and extrapolibility of the proposed mechanism.

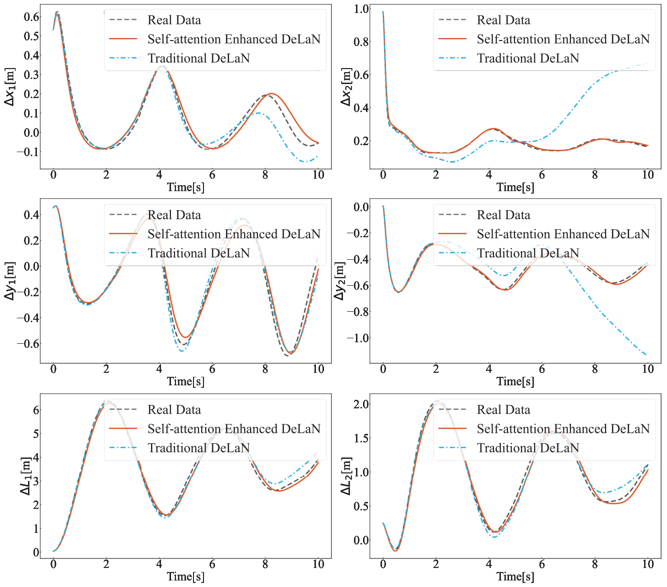


Fig. 6. The prediction performance of the proposed self-attention enhanced DeLaN.

To quantitatively evaluate performance of the proposed method, the system states predicted by the learned dynamics $q_{pred}(t)$ are compared to the data set $q_{real}(t)$. The discrete prediction error, which is treated as a benchmark for learning accuracy, is designed as

$$E = \frac{\sum_{k=0}^{T/\Delta t} \sum_i (q_{real,i}(k) - q_{pred,i}(k))^2 dt}{T/\Delta t} \tag{46}$$

where Σ sums all components, T is the simulation time. In this benchmark, to further validate effectiveness of the proposed method, the number of training epochs increased to 1000 for DeLaN without attention mechanism and decreased to 500 for attention-enhanced DeLaN. Assuming that the initial configuration is $q_0 = [0.53, 0.45, 0.05, 0.98, 0.005, 0.25]^{\mathrm{T}}$ with its first-order time derivative $\dot{q}_0 = [0.01, 0.1, 0.01, 0.1, 0.1, 0.02]^{\mathrm{T}}$, the control input $u(t) = [0.99, 0.533, 0.861, 0.814, 0.332, 0.565]^{\mathrm{T}}$ Nm is constant, and the simulation time T is set to 10 seconds.

Fig. 6 depicts the actual and predicted curves of the configuration variables of the open-loop prediction benchmark, where the advantage of the attention mechanism is clearly demonstrated. It can be observed that the configuration variables without the attention mechanism exhibit a significant divergent trend, particularly for Δx_2 and Δy_2, with prediction error $E_1 = 0.2077$. The configuration variables predicted by the attention-enhanced DeLaN are significantly more accurate than those by the model without attention mechanism, and the prediction error is $E_2 = 0.0913$, which is approximately 44% of E_1.

For a guaranteed optimal performance when applying the proposed self-attention enhanced DeLaN, hyperparameters should be chosen properly. Increasing the depth of each network, d_l, d_d, d_i, helps capture complex patterns and features in data, especially for complex and highly nonlinear systems.

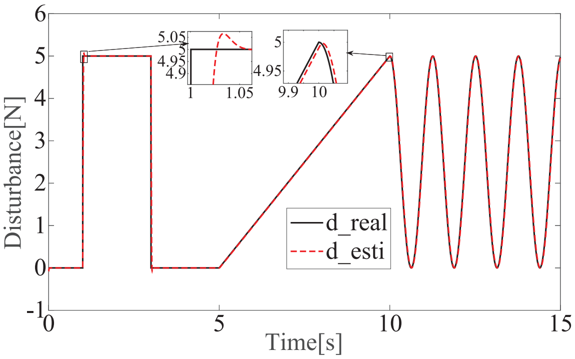


Fig. 7. Performance of the NDOB. The black solid line represents lumped uncertainties including step, linear and sine signals, while the red dashed line denotes the estimation.

However, an excessively deep network can result in gradient vanishing or exploding problems, making convergence difficult. Increasing the width of networks, w_l, w_d, w_i, can enhance network's ability to handle complex input data, while too many neurons in a layer can bring about overfitting and low computational efficiency. Learning rate is also an important hyperparameter. Choosing the learning rate η overly high can make the network diverge, and adjusting it too low may result in a low convergence rate and a lengthy learning time.

B. Tracking Performance of the Proposed Controller

The proposed controller (27) (Experimental Group, EG) is compared with the controller based on traditional DeLaN (Control Group 1, CG1), controller with fixed parameters (CG2) and an integer-order sliding mode controller (CG3). Additionally, two state-of-the-art control methods, MPC controller with data-driven deep stochastic Koopman uncertainty compensator [34] (State-of-the-art Method 1, SM1) and hybrid ALCF-MLNN model-based feedback controller [35] (SM2) are also used for comparison. The control input of CG1 and CG2 are expressed by (27), but the dynamics (*i.e.* L, $D(q)$ and $A(q)$) are learned by traditional DeLaN, k_3 and k_6 become constant for CG2. For CG3, the controller is as follows

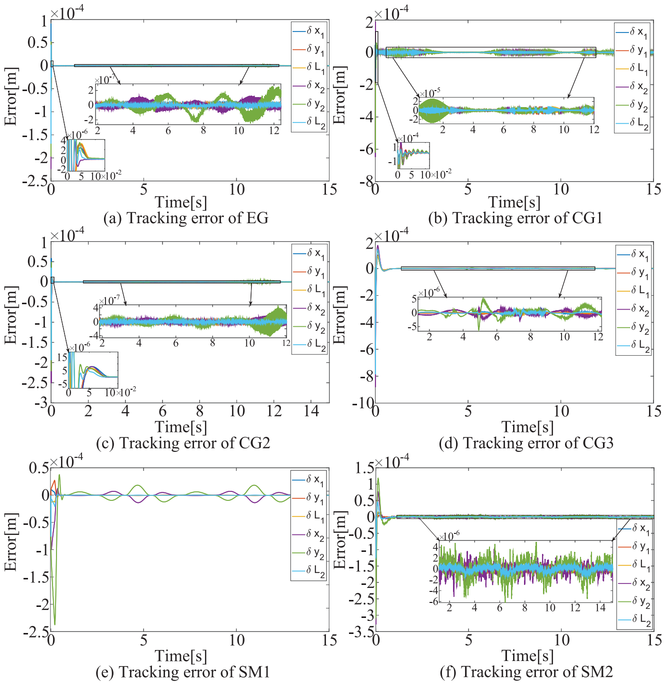
$$u = (A(q))^{-1}\Big(\frac{\partial^2 L}{\partial \dot{q}^2}\Big(\ddot{q}_{ref} + \sum_{i=1}^{r}(\mathcal{C}_r^i \lambda^i D^{2-i} e) + c_1 \mathrm{sgn}(s) + c_2 s + c_3 \mathrm{sig}^{\rho}(s)\Big) + \frac{\partial^2 L}{\partial q \partial \dot{q}}\dot{q} + D(q)\dot{q} - \frac{\partial L}{\partial q}\Big) - \delta\hat{\mathcal{F}} \quad (47)$$

in which $\mathcal{C}_r^i = \frac{i!(r-i)!}{r!}$, $s = \frac{(D+\lambda)^r}{D^{r-1}}e$. D is the differential operator. λ is a positive real number, $r \in \mathbb{N}^+$.

System parameters, the control target and controller parameters are listed in Table I and Table III, respectively. In the simulations, each generalized torque is constrained within $[-10, 10]$ Nm, the time step is set as 5×10^{-4}s. The simulation results are depicted in Figs. 7–10.

Figure 7 presents the NDOB performance in estimating lump uncertainties. From Fig. 7, the proposed NDOB can quickly estimate the system uncertainty (within 0.03 s) with a tiny overshoot (around 1.2%). Fig. 8 compares the tracking error of configuration variables under four controllers. EG and CG2 exhibit low error than CG1 and CG3, with EG converging faster (0.04 s vs. 0.08 s for CG2). In contrast, SM1 and SM2 show much higher tracking errors and slower convergence, with SM2 displaying notable oscillations. These results demonstrate that the self-attention mechanism enhances dynamics learning and the fractional-order controller improves tracking accuracy and robustness. The L_2-norm curves of the tracking errors of the controllers are shown in Fig. 9. Obviously, SM2 exhibits the largest overall error and oscillation, followed by CG1 and CG3. While EG and CG2 have similar L_2-norm error curves, EG shows less oscillation and strong disturbance rejection. The average L_2-norm values during convergence ($t \in [0, 2]$s), steady-state stage ($t \in [5, 15]$s), and the whole 15s tracking process is listed in Table IV.

Figure 10 plots the control inputs for all four controllers. SM2 and CG3 exhibit due to the frequent switching in integer-order SMC and sensitivity to high-frequency noise



(a) Tracking error of EG

(b) Tracking error of CG1

(c) Tracking error of CG2

(d) Tracking error of CG3

(e) Tracking error of SM1

(f) Tracking error of SM2

Fig. 8. Tracking error curves of different control approaches.

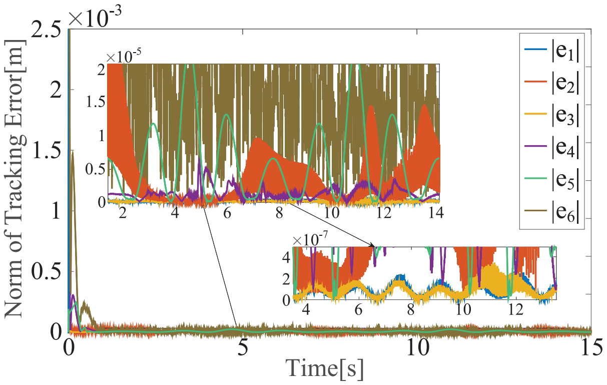
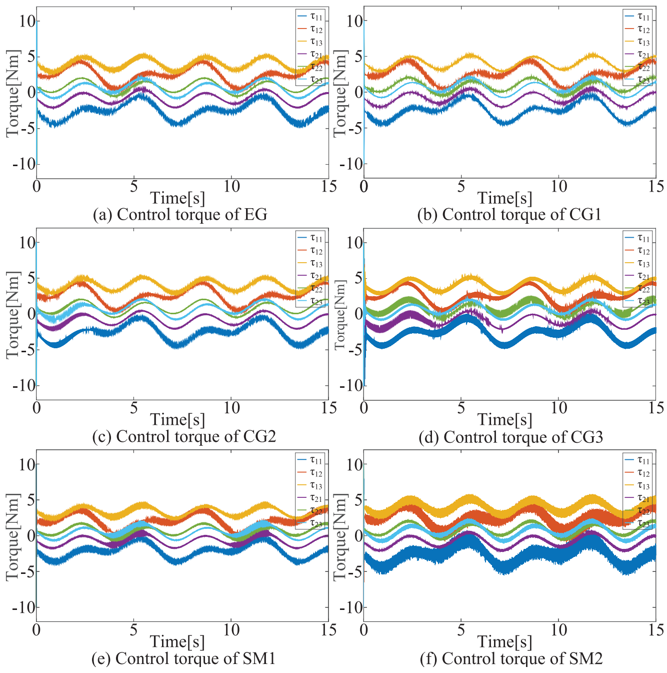


Fig. 9. L_2-norm of tracking error for different controllers. The legend $|e_1|$ to $|e_6|$ represents tracking error of controllers in the order shown in Fig. 8(a) to Fig. 8(f).



(a) Control torque of EG

(b) Control torque of CG1

(c) Control torque of CG2

(d) Control torque of CG3

(e) Control torque of SM1

(f) Control torque of SM2

Fig. 10. Control torque inputs under different controllers. τ_{ij} denotes the applied torque of j-th chamber of Segment i.

TABLE III
Control Target and Controller Parameters

Parameters of control target	**Value**
Desired configuration vector q_{ref}	$[1,1,1,1,1,1]^{\mathrm{T}} \cdot (0.2\sin t + 0.4)$
Initial configuration vector q_0	$[1,1,1,1,1,1]^{\mathrm{T}} \cdot 0.4$
Initial configuration derivative $\dot{q}_0$	$[1,1,1,1,1,1]^{\mathrm{T}} \cdot 0.2$
Simulation time T_{sim}/s	15
Disturbance amplitude and period	1, 2πs
Integer-order controller	**Value**
$\lambda, r, \rho, c_1, c_2, c_3$	10, 3, 0.5, 30, 40, 30
Fractional-order controller	**Value**
$k_1, k_2, k_3, k_4, k_5, k_6$	5, 5, 100, 10, 40, 175
$k_7, k_8, \epsilon, \beta, \nu$	2, 0.01, 0.5, 0.5, 0.5
$\rho, \gamma, \varepsilon, \alpha$	0.5, 0.5, 0.01, 0.5
Controller adaptive law	**Value**
$w_1, w_2, \varepsilon_1, \varepsilon_2$	20000, 10, 3×10^{-3}, 2.2×10^{-4}
$k_{3min}, k_{3max}, k_{6min}, k_{6max}$	50, 150, 100, 300
Parameters of NDOB	**Value**
Λ, r, a_1, α	$5I_6$, 1, 1, 0.97
DeSKO-MPC [34]	**Value**
Data points	Same as ours
Batch size	200
Network depth and width	30,4
Learning rate	1×10^{-3}
Weight decay	1.5×10^{-5}
ALCF-MLNN [35]	**Value**
Data points	Same as ours
Batch size	200
Other Parameters of MLNN	Same as [35]
K_P, K_D	25, 10

TABLE IV
L_2-Norm of Tracking Error Under Different Controllers

	$[0,2]$s	$[5,15]$s	**Whole process**
EG	1.2465×10^{-6}	7.3268×10^{-8}	2.3910×10^{-7}
CG1	6.9050×10^{-5}	1.3120×10^{-6}	1.0307×10^{-5}
CG2	2.6534×10^{-6}	8.9103×10^{-8}	4.4952×10^{-7}
CG3	1.1862×10^{-5}	5.8317×10^{-7}	2.1851×10^{-6}
SM1	3.5414×10^{-5}	5.5806×10^{-6}	9.9607×10^{-6}
SM2	2.1416×10^{-4}	1.9094×10^{-5}	4.3530×10^{-5}

TABLE V
Time Efficiency of Dynamics Learning Networks and Controllers

Dynamics learning network	**Time consumption**
Self-attention DeLaN	5978.6/11.9572
DeLaN	9463.0/9.4630
ANCF-MLNN [35]	11472.5/11.4725
Controller	**Time consumption**
EG	$168.2/5.627 \times 10^{-3}$
CG2	$167.6/5.587 \times 10^{-3}$
CG3	$142.5/4.750 \times 10^{-3}$
SM1 [34]	$259.1/8.637 \times 10^{-3}$
SM2 [35]	$155.0/5.167 \times 10^{-3}$

in PID-like controllers, while CG2 experiences occasional oscillations. In contrast, EG, CG1, and SM1 show minimal oscillation, benefiting from the filtering effect and strong high-frequency uncertainty suppression of the fractional-order controller. Additionally, the MPC method in SM1 helps reduce oscillations. EG demonstrates the highest stability in control torque. In practical applications, the chamber pressure rate of change is limited, making sudden changes impractical, further highlighting advantages of the proposed controller.

C. Computation Efficiency

The computation efficiency is critical for implementing model learning algorithms and control methods in real-world applications and real-time control. This subsection discusses the computational cost of the self-attention-enhanced DeLaN and the proposed controller. As shown in Table V, we recorded the time consumption for dynamics learning and trajectory tracking simulations of different control methods. The total time (in seconds) is listed on the left of the "Time consumption" column, while the average time per epoch/time step is shown on the right. All simulations were performed on a laptop with an Intel Core i9-13900 CPU. Note that DeSKO-MPC [34] is a online control method for estimating model uncertainties and predicting future system states, and we focus only on its control efficiency.

From Table V, the proposed self-attention DeLaN has the lowest overall learning time as the self-attention mechanism accelerates convergence, though it incurs a slightly higher time cost per epoch. SM1 has the highest time consumption due to the online neural network and long-term MPC controller, while CG3 requires the least time due to its simpler design. Although the proposed controller is somewhat more computationally

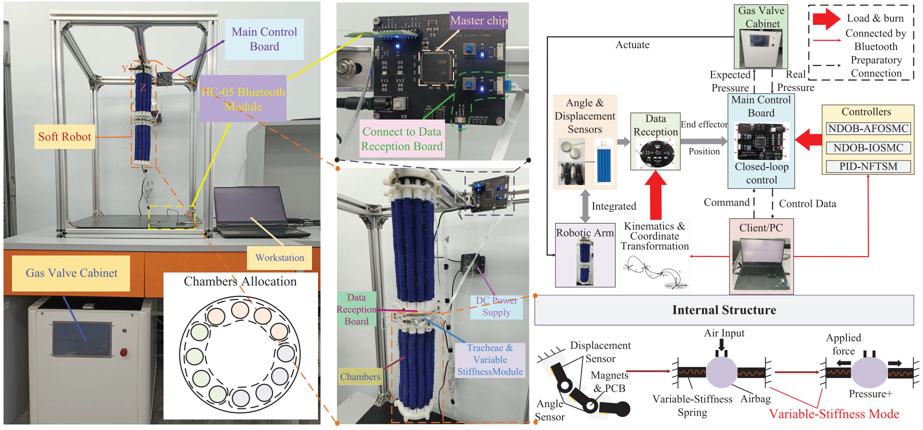


Fig. 11. The soft robotic experimental platform. The left image shows the overview, highlighting the main components, as well as the chamber distribution. The upper-middle diagram depicts the structure of the main control board, which carries the control algorithm and calculates the real-time input pressure. The pneumatically actuated soft arm is demonstrated in the lower-middle figure, operates through adjustable pressure outputs controlled by the gas valve cabinet. The right diagram shows the communication and connection relationship of the entire platform and the internal structure of each segment of the soft arm.

TABLE VI

CONTROLLER PARAMETERS IN EXPERIMENTS

NDOB-AFOSMC	**Value**
$k_1, k_2, k_3, k_4, k_5, k_6$ $k_7, k_8, \epsilon, \beta, \nu$ $\rho, \gamma, \varepsilon, \alpha$	5, 2, 10, 20, 50, 100 2.5, 1, 0.2, 0.5, 0.5 0.5, 0.5, 0.2, 0.5
NDOB-IOSMC	**Value**
$\lambda, r, \rho, c_1, c_2, c_3$	3, 3, 0.5, 3, 10, 8
PID-NFTSM [38]	**Value**
$k_1, k_2, k_{3,max}, k_{3,min}$ $k_p, k_i, k_d, \eta_1, \eta_2$ $\rho_1, \rho_2, \varepsilon, \Theta, \phi, L$	$2I_3, I_3, 0.5I_3, 4I_3$ $4I_3, 0.4I_3, 1.6I_3, I_3, I_3$ $0.9I_3, 1.1I_3, 0.2I_3, 2I_3, 0.01I_3, 0.03s$
Parameters of NDOB	**Value**
Λ, r, a_1, α	$5I_3, 1, 1, 0.6$
DeSKO-MPC [34]	**Value**
Batch size Network depth and width Learning rate Weight decay	100 20,3 2.5×10^{-3} 5×10^{-5}

TABLE VII

MEAN EUCLIDEAN TRACKING ERROR (MM) UNDER DIFFERENT CONTROLLERS AND TRAJECTORIES

Trajectory A	**Stage 1**	**Stage 2**	**Whole process**
NDOB-AFOSMC	14.538	1.877	2.780
NDOB-IOSMC	18.021	2.226	3.352
PID-NFTSM	20.984	2.269	3.603
DeSKO-MPC	16.286	2.363	3.356
Trajectory B	**Stage 1**	**Stage 2**	**Whole process**
NDOB-AFOSMC	7.071	2.124	2.476
NDOB-IOSMC	9.683	2.736	3.232
PID-NFTSM	12.684	3.180	3.858
DeSKO-MPC	10.806	2.726	3.303
Trajectory C	**Stage 1**	**Stage 2**	**Whole process**
NDOB-AFOSMC	13.317	2.327	3.111
NDOB-IOSMC	16.292	3.165	4.101
PID-NFTSM	18.992	3.302	4.421
DeSKO-MPC	17.625	2.637	3.706

expensive than others (due to the integer-order approximation for fractional-order calculus [33]), it significantly outperforms in tracking performance. To further optimize computational costs, simplification methods can be applied, such as using an adaptive connection policy to disconnect “unimportant” nodes with low attention scores, reducing the backpropagation cost.

V. EXPERIMENTS

Experiments are conducted on a two-segments continuous pneumatic soft manipulator (shown in Fig. 11). Each segment is actuated by 12 chambers grouped in three (every 4 chambers form a group), with independent pressure input for each group. The soft part of each segment measures 208 mm in length and 75 mm in diameter, with chamber diameter of 15 mm and segment weight of 00.6 kg. Multiple linked angle and displacement sensors are embedded inside the robot to measure the configuration states. The end-effector pose is computed via multi-joint kinematics as in [36], which is integrated in the data acquisition board. The Young’s modulus is set to 2×10^4 Pa through its variable stiffness module. The nominal dynamics is modeled using the self-attention enhanced DeLaN. Configuration to task space transformation follows [32], and the chamber pressure to torque mapping uses the method in [37].

In the experiments, only the second segment is actuated. The deformation of this segment induces coupled motion in the first, which acts as an external disturbance, increasing system uncertainty. Each trial lasts 224.4 s, with control and data acquisition executed at 20 Hz. To prevent overpressure, each chamber group’s pressure p_i is limited to $p_i \in [0, 1.5]$ Bar. Since the fractional-order calculus in Definitions 1- 2 is complex and difficult to implement, a reduced-length Grünwald–Letnikov form ([39], Def. 1) is applied, with $\alpha = 0.5$, $L = 6$ and $h = 0.05$. The reference

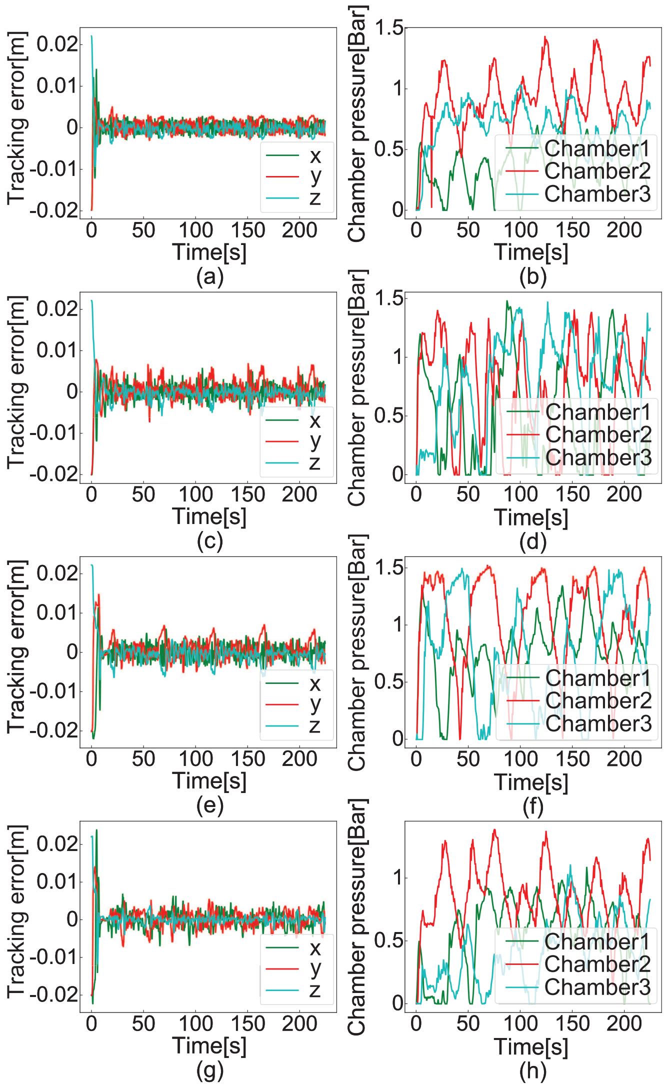


Fig. 12. Tracking error and chamber pressure for trajectory A under NDOB-AFOSMC (Figs. (a)-(b)), NDOB-IOSMC (Figs. (c)-(d)), PID-NFTSM (Figs. (e)-(f)) and DeSKO-MPC (Figs. (g)-(h)).

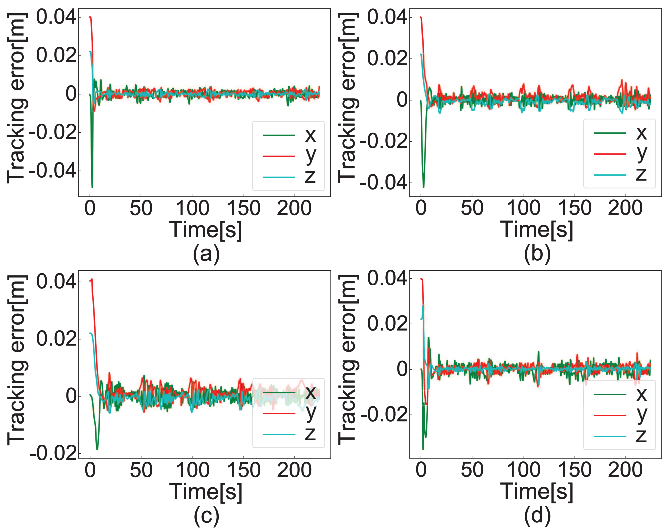


Fig. 13. Tracking error for trajectory B under NDOB-AFOSMC (Fig. (a)), NDOB-IOSMC (Fig. (b)), PID-NFTSM (Fig. (c)) and DeSKO-MPC (Fig. (d)).

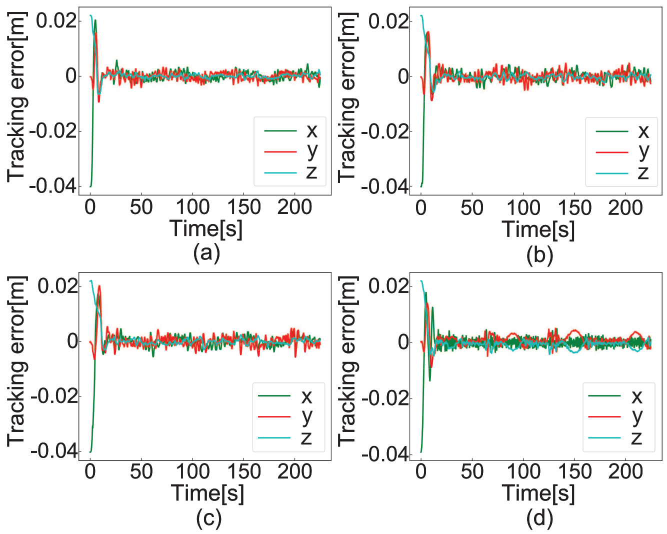


Fig. 14. Tracking error for trajectory C under NDOB-AFOSMC (Fig. (a)), NDOB-IOSMC (Fig. (b)), PID-NFTSM (Fig. (c)) and DeSKO-MPC (Fig. (d)).

trajectories are defined as: A) a butterfly-shaped (figure- 8) path composed of two semi-circles (radius 20 mm) and two straight segments, B) an equilateral triangle, and C) a circle. All three trajectories lie on the plane $z = 230$ mm and share a circumradius of 40 mm. The soft robot is commanded to follow a two-stage task-space reference signal q_{ref}: a static convergence phase to $q_{ref} = [-40, 0, 230]^{\mathrm{T}}$mm in the first 16 seconds (Stage 1), followed by dynamic trajectory tracking for the next 208.4 seconds (Stage 2). The motion period is set to 60s for trajectory A and 48 s for trajectories B and C. During experiments, the task-space trajectories are mapped into the configuration space. The proposed control scheme (NDOB-AFOSMC) is compared to NDOB based integer-order sliding mode control (NDOB-IOSMC), PID-NFTSM [38] and DeSKO-MPC [34]. Table VI lists the parameters of the four controllers. Note that the disturbance observer parameters used in NDOB-IOSMC are identical to those in NDOB-AFOSMC. In PID-NFTSM, parameter $\overline{M}$ is replaced by the inverse of input transform matrix, $\bar{A}^{-1}$. To compensate for the hysteresis effect of the soft arm, a pressure-tuning strategy is adopted:let p_i be the current pressure computed by the controller for Chamber i, and p_{ki} the previous input pressure. The actual input pressure $\overline{p}_i$ is then adjusted according to (48). Experimental results are presented in Figs. 12–16.

$$\overline{p}_i = \begin{cases} p_i & p_{ki}p_i \geq 0 \\ p_i + 20\mathrm{sgn}(p_i - p_{ki}) & p_{ki}p_i < 0 \end{cases} \tag{48}$$

As shown in Figs. 12–14, all four controllers are able to regulate the robot to follow the desired trajectories. However, their performance varies significantly. NDOB-AFOSMC achieves lower tracking error and faster convergence compared to the others. It also exhibits reduced input pressure overshoot in Stage 1 and smoother pressure fluctuations (Fig. 12), indicating lower working loss and power consumption in the gas valve cabinet. Table VII compares the mean Euclidean tracking error under the four different controllers and three

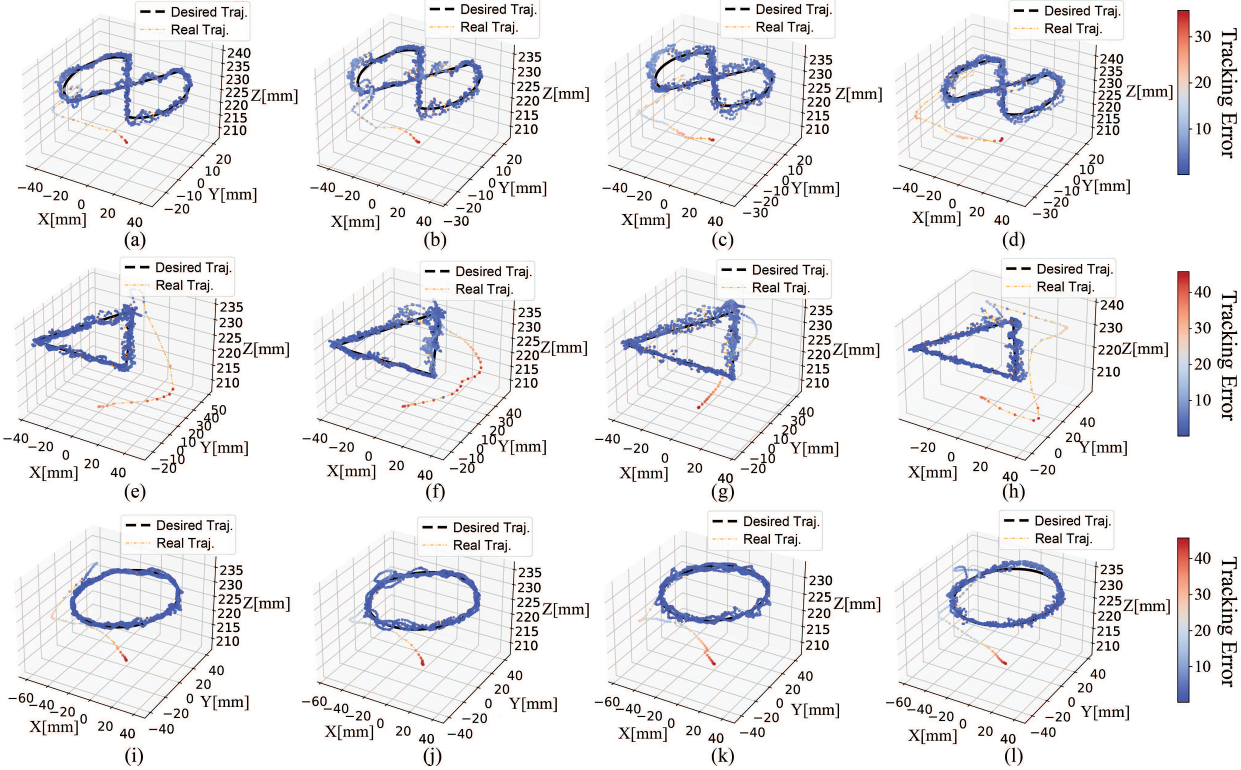


Fig. 15. End-effector trajectory under different controllers. The controllers from left to right are NDOB-AFOSMC, NDOB-IOSMC, PID-NFTSM and DeSKO-MPC. Figs. (a)-(d), (e)-(h) and (i)-(l) show the tracking performance under trajectories A, B and C, respectively.

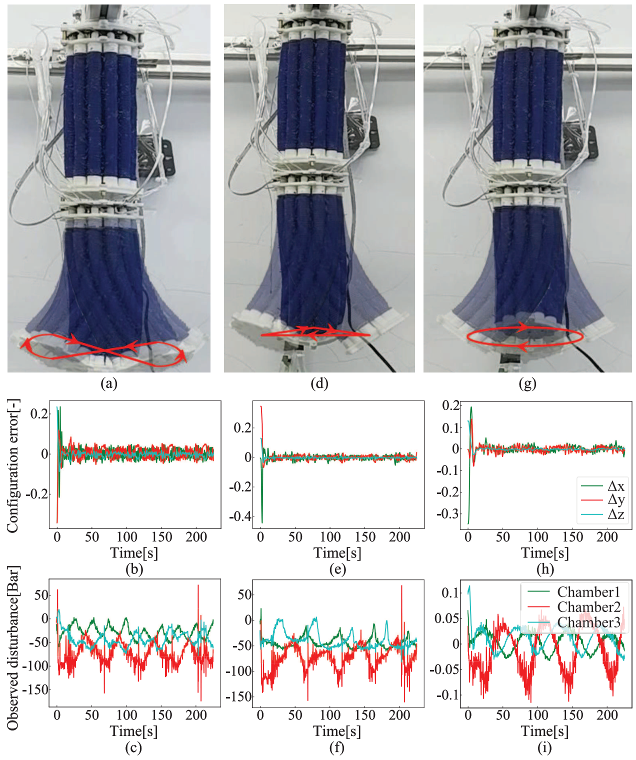


Fig. 16. System motion under the NDOB-AFOSMC for the three trajectories. (a)-(c) Motion sequences, configuration errors and observed disturbance for trajectory A. (d)-(f) for trajectory B. (g)-(i) for trajectory C.

trajectories. Fig. 15 further highlights the superior end-effector tracking of NDOB-AFOSMC across all trajectories, with minimal deviation and oscillation. In Fig. 16(a), we show the motion sequence of the soft arm under Trajectory A, while the corresponding configuration-space tracking error and estimated disturbance are presented in Figs. 16(b)–(c). Similar results for Trajectories B and C are illustrated in Figs. 16(d)-(i).

VI. Conclusion

This paper presents a self-attention enhanced DeLaN for learning the dynamics of soft robots and an adaptive fractional-order sliding mode controller (AFOSMC) with lumped disturbance compensation based on a nonlinear disturbance observer (NDOB). The proposed controller demonstrates strong robustness and achieves accurate trajectory tracking despite model uncertainties, which are effectively compensated by the NDOB. The effectiveness of the modeling and control schemes is validated through both simulations and experiments on a soft robotic platform. Future work will focus on enhancing the attention mechanism architecture, addressing input saturation, and enabling online estimation of unknown dynamic parameters.

References

[1] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of continuum and soft robots: A strain parameterization based approach," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 847–863, Jun. 2021.

[2] C. Armanini et al., "Soft robotics for farm to fork: Applications in agriculture & farming," *Bioinspiration Biomimetics*, vol. 19, no. 2, Mar. 2024, Art. no. 021002.

[3] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010.

[4] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 935–949, Aug. 2014.

[5] F. Renda, V. Cacucciolo, J. Dias, and L. Seneviratne, "Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 5495–5502.

[6] Z. Chen et al., "Data-driven methods applied to soft robot modeling and control: A review," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 2241–2256, 2025.

[7] T. Navez, E. Ménager, P. Chaillou, O. Goury, A. Kruszewski, and C. Duriez, "Modeling, embedded control and design of soft robots using a learned condensed FEM model," *IEEE Trans. Robot.*, vol. 41, pp. 2441–2459, Jan. 2025.

[8] H. Sharma and B. Kramer, "Preserving Lagrangian structure in data-driven reduced-order modeling of large-scale dynamical systems," *Phys. D, Nonlinear Phenomena*, vol. 462, Jun. 2024, Art. no. 134128.

[9] Z. K. Lawal, H. Yassin, D. T. C. Lai, and A. Che Idris, "Physics-informed neural network (PINN) evolution and beyond: A systematic literature review and bibliometric analysis," *Big Data Cognit. Comput.*, vol. 6, no. 4, p. 140, Nov. 2022.

[10] M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian networks: Using physics as model prior for deep learning," in *Proc. Int. Conf. Learn. Represent.*, Jul. 2019, pp. 1–17.

[11] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 1–11.

[12] Z. Huang, M. Liang, J. Qin, S. Zhong, and L. Lin, "Understanding self-attention mechanism via dynamical system perspective," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 1412–1422.

[13] D. Malas, S. Wang, W. Huang, L. Lindenroth, W. Xia, and H. Liu, "A novel pneudraulic actuation method to enhance soft robot control," *Soft Robot.*, vol. 11, no. 1, pp. 70–84, Jan. 2024.

[14] X. Shao et al., "Model-based control for soft robots with system uncertainties and input saturation," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 7435–7444, Jul. 2024.

[15] D. Papageorgiou, G. Siguroardóttir, E. Falotico, and S. Tolu, "Sliding-mode control of a soft robot based on data-driven sparse identification," *Control Eng. Pract.*, vol. 144, Mar. 2024, Art. no. 105836.

[16] A. M. Kizhakkethil et al., "Self-evolving fuzzy-based control of soft pneumatic rotary actuators," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2024, pp. 939–944.

[17] A. Mousavi, A. Ahmed, H. Khaksar, H. Choi, and A. Kafash Hoshiar, "An input saturation-tolerant position control method for magnetic microrobots using adaptive fuzzy sliding-mode method," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 3852–3865, 2025.

[18] Y. Wang, Y. Liu, and F. Xu, "Adaptive iterative learning control of soft robot for beating heart tracking," *Robotic Intell. Autom.*, vol. 44, no. 3, pp. 488–497, Jun. 2024.

[19] Q. Meng, X. Sun, Y. Wang, J. Wu, and C.-Y. Su, "Modeling and neural-network-based tail oscillation control of a fish-like bionic soft actuation mechanism," *IEEE Robot. Autom. Lett.*, vol. 10, no. 4, pp. 3827–3834, Apr. 2025.

[20] Z. Chen, M. Bernabei, V. Mainardi, X. Ren, G. Ciuti, and C. Stefanini, "A novel and accurate BiLSTM configuration controller for modular soft robots with module number adaptability," *Soft Robot.*, vol. 11, no. 6, pp. 1–10, Dec. 2024.

[21] X. Shao, G. Sun, W. Yao, X. Li, and O. Zhang, "Fractional-order resolved acceleration control for free-floating space manipulator with system uncertainty," *Aerosp. Sci. Technol.*, vol. 118, Nov. 2021, Art. no. 107041.

[22] J. De La Morena, D. Redrejo, F. Ramos, V. Feliu, and A. S. Vázquez, "Fractional order modeling and control of hydrogel-based soft pneumatic bending actuators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2024, pp. 11362–11367.

[23] Z. Du, L. Yang, Y. Sun, and X. Chen, "Motion control for continuum robots: A mini review for model-free and hybrid-model control," in *Proc. Int. Conf. Intell. Robot. Appl.*, Jan. 2025, pp. 372–391.

[24] S. Wu, Z. Li, W. Chen, and F. Sun, "Dynamic modeling of robotic manipulator via an augmented deep Lagrangian network," *Tsinghua Sci. Technol.*, vol. 29, no. 5, pp. 1604–1614, Oct. 2024.

[25] Z. Li, S. Wu, W. Chen, and F. Sun, "Extrapolation of physics-inspired deep networks in learning robot inverse dynamics," *Mathematics*, vol. 12, no. 16, p. 2527, Aug. 2024.

[26] C. Della Santina, "Flexible manipulators," *Encyclopedia Robot.*, vol. 20, pp. 1–15, Jan. 2021.

[27] B. Jin et al., *Fractional Differential Equations*. Cham, Switzerland: Springer, 2021.

[28] A. Erdélyi, "Fractional integrals of generalized functions," in *Proc. Int. Conf. Held at Univ. New Haven Fractional Calculus Appl.*, Jan. 1975, pp. 151–170.

[29] S. Khoo, L. Xie, S. Zhao, and Z. Man, "Multi-surface sliding control for fast finite-time leader-follower consensus with high order SISO uncertain nonlinear agents," *Int. J. Robust Nonlinear Control*, vol. 24, no. 16, pp. 2388–2404, Nov. 2014.

[30] M. Ö. Efe, "Fractional order sliding mode controller design for fractional order dynamic systems," in *New Trends in Nanotechnology and Fractional Calculus Applications*. Berlin, Germany: Springer, 2010, pp. 463–470.

[31] M. Chen, G. Tao, and B. Jiang, "Dynamic surface control using neural networks for a class of uncertain nonlinear systems with input saturation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2086–2097, Sep. 2015.

[32] C. D. Santina, A. Bicchi, and D. Rus, "On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1001–1008, Apr. 2020.

[33] A. Oustaloup, F. Levron, B. Mathieu, and F. M. Nanot, "Frequency-band complex noninteger differentiator: Characterization and synthesis," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 47, no. 1, pp. 25–39, Jan. 2000.

[34] M. Han, K. Wong, J. Euler-Rolle, L. Zhang, and R. K. Katzschmann, "Robust learning-based control for uncertain nonlinear systems with validation on a soft robot," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 510–524, Jan. 2025.

[35] X. Huang, Y. Rong, and G. Gu, "High-precision dynamic control of soft robots with the physics-learning hybrid modeling approach," *IEEE/ASME Trans. Mechatronics*, vol. 30, no. 3, pp. 1658–1669, Jun. 2025.

[36] X. Shao, G. Sun, C. Xue, and X. Li, "Nonsingular terminal sliding mode control for free-floating space manipulator with disturbance," *Acta Astronautica*, vol. 181, pp. 396–404, Apr. 2021.

[37] C. D. Santina, A. Bicchi, and D. Rus, "Dynamic control of soft robots with internal constraints in the presence of obstacles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6622–6629.

[38] Y. Wang, L. Liu, M. Yuan, Q. Di, B. Chen, and H. Wu, "A new model-free robust adaptive control of cable-driven robots," *Int. J. Control, Autom. Syst.*, vol. 19, no. 9, pp. 3209–3222, Sep. 2021.

[39] G. Sun, L. Wu, Z. Kuang, Z. Ma, and J. Liu, "Practical tracking control of linear motor via fractional-order sliding mode," *Automatica*, vol. 94, pp. 221–235, Aug. 2018.

Xiangyu Shao (Member, IEEE) received the B.S. degree in automation from Harbin Engineering University, Harbin, China, in 2016, and the M.S. and Ph.D. degrees in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2018 and 2022, respectively.

He is currently an Associate Professor with the School of Astronautics, Harbin Institute of Technology. His research interests include space robots, soft robots, sliding mode control, and fractional order control.

Linke Xu (Graduate Student Member, IEEE) received the B.S. degree in automation from Harbin Institute of Technology, Harbin, China, in 2024, where he is currently pursuing the Ph.D. degree in control science and engineering.

His research interests include soft robots, sliding mode control, fractional-order control, and learning-based dynamics modeling.

Guanghui Sun (Senior Member, IEEE) received the B.S. degree in automation and the M.S. and Ph.D. degrees in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2005, 2007, and 2010, respectively.

He is currently a Professor with the Department of Control Science and Engineering, Harbin Institute of Technology. His research interests include fractional-order systems, networked control systems, and sliding mode control.

Weiran Yao (Member, IEEE) received the bachelor's (hons.) degree, the master's degree, and the Ph.D. degree in aeronautical and astronautical science and technology from the School of Astronautics, Harbin Institute of Technology (HIT), Harbin, China, in 2013, 2015, and 2020, respectively. From 2017 to 2018, he was a Visiting Ph.D. student with the Department of Mechanical and Industrial Engineering, University of Toronto (UofT), Toronto, Canada. He is currently an Associate Professor with the School of Astronautics, HIT. His research interests include autonomous vehicles, multi-robot mission planning, and multi-agent control systems.

Ligang Wu (Fellow, IEEE) received the B.S. degree in automation from Harbin University of Science and Technology, China, in 2001, the M.E. degree in navigation guidance and control from Harbin Institute of Technology, China, in 2003, and the Ph.D. degree in control theory and control engineering from Harbin Institute of Technology, in 2006. From January 2006 to April 2007, he was a Research Associate with the Department of Mechanical Engineering, The University of Hong Kong, Hong Kong. From September 2007 to June 2008, he was a Senior Research Associate with the Department of Mathematics, City University of Hong Kong, Hong Kong. From December 2012 to December 2013, he was a Research Associate with the Department of Electrical and Electronic Engineering, Imperial College London, London, U.K. In 2008, he joined Harbin Institute of Technology, China, as an Associate Professor, and was then promoted to a Full Professor in 2012. He has published seven research monographs and more than 170 research articles in internationally refereed journals. His current research interests include switched systems, stochastic systems, computational and intelligent systems, sliding mode control, and advanced control techniques for power electronic systems.

He was the winner of the National Science Fund for Distinguished Young Scholars in 2015 and received China Young Five Four Medal in 2016. He was named as the Distinguished Professor of Chang Jiang Scholar in 2017, and was named as the Highly Cited Researcher in 2015 to 2019. He currently serves as an Associate Editor for several journals, including IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, *Information Sciences, Signal Processing*, and *IET Control Theory and Applications*. He is also an Associate Editor for the Conference Editorial Board, IEEE Control Systems Society.

Cosimo Della Santina (Senior Member, IEEE) received the Ph.D. degree in robotics from the University of Pisa, Italy, in 2019. He was a Visiting Ph.D. Student and a Postdoctoral Researcher with Massachusetts Institute of Technology, USA, from 2017 to 2019. He was a senior post-doctoral researcher and a Guest Lecturer with the Department of Informatics, Technical University of Munich (TUM), from 2020 to 2021. Since 2020, he has been affiliated with German Aerospace Centre (DLR) as an external research scientist. He is currently an Associate Professor with the Department of Cognitive Robotics, Delft University of Technology, The Netherlands. His research interests include motor intelligence of physical systems, focusing on mechanical systems, high dimensional dynamics, and soft robots.