Learning-Based Tracking Control of Soft Robots

Jingting Zhang, Xiaotian Chen, Paolo Stegagno, *Member, IEEE*, Mingxi Zhou, *Member, IEEE*, and Chengzhi Yuan, *Member, IEEE*

***Abstract*—This letter proposes an adaptive radial basis function neural network (RBF NN) based scheme for the dynamics learning and tracking control problems of a soft trunk robot. Specifically, a low-order approximate model describing the soft robot's dynamics is first derived with the finite element method and proper orthogonal decomposition technique. Based on this model, an adaptive learning control scheme is developed with RBF NN, which can not only provide stable and accurate tracking control for the soft robot, but also achieve accurate learning of the robot's dynamics during the online control process. The proposed controller can effectively handle the soft robot's complex nonlinear uncertain dynamics and external disturbances, it thus can guarantee desirable tracking accuracy and control adaptability. The learned knowledge of robot's dynamics can be obtained and stored in a constant RBF NN model. Based on this, a novel knowledge-based controller is further proposed to provide desirable control performance for the soft robot without needing to repeat any online parameter adaptations, which significantly improves the overall system's operational efficiency with reduced computational complexity and easier control implementation. Effectiveness and advantages of the proposed methods are validated through physical experiments.**

***Index Terms*—Soft robotics, tracking control, dynamics learning, adaptive learning control, neural networks.**

I. Introduction

SOFT robots are a novel type of robots made of soft materials, such as silicone and robber. Compared to traditional rigid robots, soft robots have developed many desirable mechanical properties, e.g., inherent compliance, flexibility and hyper redundance, which facilitates safe human-robot interaction and operation in a restrained environment [1]. This has motivated a rapidly-increasing demand of soft robots for industrial, surgical and assistive applications [2].

Manuscript received 30 March 2023; accepted 19 July 2023. Date of publication 9 August 2023; date of current version 18 August 2023. This letter was recommended for publication by Associate Editor V. Modugno and Editor J. Kober upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation under Grant CMMI-1929729. *(Corresponding author: Paolo Stegagno.)*

Jingting Zhang is with the Center for Robotics, School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: zhangtingzi95@163.com).

Xiaotian Chen and Chengzhi Yuan are with the Department of Mechanical, Industrial and Systems Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: xiaotian_chen@uri.edu; cyuan@uri.edu).

Paolo Stegagno is with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: pstegagno@uri.edu).

Mingxi Zhou is with the Graduate School of Oceanography, University of Rhode Island, Narragansett, RI 02882 USA (e-mail: mzhou@uri.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3303724, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3303724

Precise motion control of soft robots is an essential and difficult problem. The main technical challenge lies in the soft robot's complex physical model due to its deformable structure, complex geometry and infinite degrees of freedom [2]. To this end, [20] used the Geometric Variable Strain technique—a geometrically exact approach based on the Cosserat rod theory—to model the dynamics of the soft robot. [21] presented the deformation space formulation of the soft robots' dynamics by using the Finite Element Deformation Method. [3] used the piecewise constant curvature approximation technique for kinematics modeling of soft robots. In [4], Euler-Bernoulli Beam theory has been used to model the bending of robots. These schemes require strict assumptions on the shape, structure, and constitutive materials of the studied robots, limiting their applicability to more general and complex soft robots. Some research works in [22], [23] studied the control problem by physically modeling the soft robots without requiring strict assumptions. These methods may not be able to deal with the uncertainty or disturbances of the soft robots when operating in complex and variable environments.

To tackle the challenges in deriving an accurate kinematic/dynamic model for soft robots, research attempts have been dedicated to learning control design for soft robots [5]. These methods only use measurement data of soft robots, without needing *a priori* knowledge of robots' underlying structure. They can develop arbitrarily complex kinematic/dynamic models of soft robots, which are especially suitable for those robots that are highly nonlinear, nonuniform, and/or act within unstructured environment [2]. Usually, learning control design are developed using neural network (NN) techniques with remarkable approximation and learning capabilities. For example, [24] utilized feedforward NN to learn a differentiable model of a soft robot's quasi-static physics, aiming to find optimal open-loop control inputs. In [6], a deep learning-based predictive uncertainty estimation framework was proposed with recurrent NN technique for soft robot's multimodal-sensing. [7], [8], [9] proposed a model-based policy learning algorithm with recurrent NN for the closed-loop predictive control problem of a soft robotic manipulator. These methods were mostly offline-learning schemes, which cannot handle abrupt changes in robot's kinematics/dynamics during online operation under various harsh control environments. To overcome these issues, some control schemes have been developed in [25], [26], which can first offline-learn the robots' model and then online-adapt to handle the disturbances or dynamics change. However, they cannot realize accurate identification for the dynamics of the soft robot while guaranteeing the associated parameter convergence

to the optimal values, thus they may not be able to achieve satisfactory sample efficiency and desired control performance.

In real applications, soft robots usually encounter external disturbances, such as payload and external interaction [5], possibly leading to major changes in robots' kinematics/dynamics. To achieve a desirable control performance in these cases, online learning control approach is needed to adapt/react to such changes by adjusting associated control variables online, see, e.g., [10], [11]. However, the research of online learning control of soft robots is still in its primitive stage with limited success so far. To the author's best knowledge, there are only a few research results have been developed in [5], [12], [13], which provided enhanced control accuracy and adaptability for soft robots in complex environments. However, these methods have not well explored the real learning capability of the associated controllers. More specific, these methods cannot obtain the learned knowledge through the control process for reutilization, which thus have to repeat online updating control parameters even for a same/similar control task, resulting in high computation burden and time consumption in associated control implementations.

In this letter, we will propose an adaptive NN learning-based tracking control scheme for a soft trunk robot, aiming to drive the robot's end-effector to track a prescribed reference trajectory with a desirable tracking accuracy. Specifically, for the soft robot, a low-order approximate model to describe the robot's dynamics will be first derived with the Finite Element Method (FEM) and Proper Orthogonal Decomposition (POD) techniques. Based on this analytical model, an adaptive learning control scheme will be proposed with the Radial Basis Function Neural Network (RBF NN) technique, which is able i) to provide accurate and stable tracking control for the soft robot's end-effector; and ii) to achieve accurate learning for the robot's uncertain dynamics during online control process. With these features, the proposed learning controller can handle the soft robot's complex uncertain dynamics and external disturbances, thus guaranteeing desirable tracking accuracy and control adaptability. Moreover, this learning controller guarantees that the knowledge of the robot's dynamics can be learned in real time during online control process, and the learned knowledge can be further stored and represented by a constant RBF NN model. Based on this, a new knowledge-based controller will be further proposed to provide a desired control action using the learned knowledge, which does not need to repeat online the update of control parameters, thereby improving the overall system's operational efficiency with reduced computational complexity and easier control implementation. Effectiveness and advantages of the proposed methods will be validated through physical experiments.

The major contributions of this work are summarized below.

i) We propose an adaptive RBF NN-based learning control scheme for soft robots, which can provide desired tracking control performance by online learning the robot's complex nonlinear uncertain dynamics;
ii) We propose a novel knowledge-based controller for soft robots, which can achieve high tracking control performance with the learned knowledge of robot's dynamics, significantly improving the overall system's operational efficiency;
iii) We perform physical experiments on a soft trunk robot to validate the effectiveness and advantages of our proposed control methods.

It should be stressed that the present work is developed by extending our previous work in [14], which focused on the dynamics modeling problem of soft robots. Different from [14], which presented an offline learning scheme, the current letter proposes an online learning scheme, which guarantees that the robot's dynamics is learned/modeled during the online control process, and the learned knowledge can be used in real time to improve the control performance. Compared to [14], the online dynamics learning mechanism in the current letter enables the controller to develop improved tracking accuracy and control adaptability, which is more suitable for soft robots operating in harsh control environments.

The remainder of this letter is organized as follows. Section II provides some preliminaries. Section III includes the problem statement. The proposed adaptive NN-based learning control scheme and the knowledge-based control scheme are presented in Section IV. The experiments are conducted in Section V. The conclusions are given in Section VI.

II. Preliminary

A radial basis function neural network (RBF NN) [15] can be described by $f_{nn}(x) = \sum_{i=1}^{N_n} \hat{w}_i s_i(x) = \hat{W}^\top S(x)$, where $x \in \Omega_x \subset \mathbb{R}^n$ is the input vector with Ω_x being a compact set, $\hat{W} = [\hat{w}_1, \ldots, \hat{w}_{N_n}]^\top \in \mathbb{R}^{N_n}$ is the weight vector, with N_n denoting the NN node number, and $S(x) = [s_1(\|x - \varsigma_1\|), \ldots, s_{N_n}(\|x - \varsigma_{N_n}\|)]^\top : \mathbb{R}^n \to \mathbb{R}^{N_n}$, with $s_i(\cdot)$ being a radial basis function, and $\varsigma_i \in \mathbb{R}^n$ $(i = 1, 2, \ldots, N_n)$ being distinct points in state space. In this paper, the radial basis function $s_i(\cdot)$ is chosen as the Gaussian function: $s_i(\|x - \varsigma_i\|) = \exp[\frac{-(x-\varsigma_i)^\top (x-\varsigma_i)}{\eta_i^2}]$, where ς_i is the center of the receptive field and η_i is the width of the receptive field. As shown in [15], for any continuous function $f(x) : \Omega_x \to \mathbb{R}$, and for the NN approximator with the node number N_n being sufficiently large, there exists an ideal constant weight vector $W^* \in \mathbb{R}^{N_n}$, such that for any $\epsilon^* > 0$, $f(x) = W^{*\top} S(x) + \epsilon, \forall x \in \Omega_x$, where $|\epsilon| < \epsilon^*$ is the ideal approximation error. The ideal weight vector W^* is an "artificial" quantity required for analysis, and is defined as the value of $\hat{W}$ that minimizes $|\epsilon|$ for all $x \in \Omega_x$, i.e., $W^* := arg\min_{\hat{W} \in \mathbb{R}^{N_n}} \{\sup_{x \in \Omega_x} |f(x) - \hat{W}^\top S(x)|\}$.

III. Problem Statement

A. The Soft Trunk Robot

The studied soft trunk robot is shown in Fig. 1, which is similar to the one in our previous work [14]. It is pneumatic actuated and composed of three identical segments made by high elasticity silicone rubber. Each segment has maximum length of 108 mm and maximum width of 32 mm. They can extend and shrink vertically by pressurizing/depressurizing the air inside the segment, and the maximum deformation displacement is about 60 mm. With the cooperative deformation of these segments, the

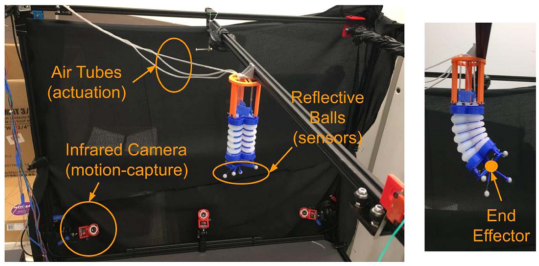


Fig. 1. Soft trunk robot and the experimental platform.

robot's end-effector can move in the 3D space. The objective of this letter is to design a controller to drive the robot's end-effector to track a given reference trajectory with a desirable tracking accuracy.

Remark 1: We emphasize that the soft trunk robot of Fig. 1 is used to provide a simple case study to validate the feasibility, effectiveness and applicability of our approach. Our approach is not necessarily limited to the robot of Fig. 1, and it can be easily extended and applied to generic soft robots.

B. Model Description

We first derive an analytical model for the robot in Fig. 1 to describe its complex geometry and continuously deforming structure. With the FEM technique, we discretize the robot's structure into a mesh of finite elements, so as to establish a finite element model as seen in Fig. 1. Denoting n as the number of the mesh nodes, we can define $q \in \mathbb{R}^{3n}$ as the 3D-displacement of each mesh node, and $v \in \mathbb{R}^{3n}$ as the velocity vector. By Newton's second law, the robot's motion can be described by the following nonlinear dynamical model:

$$M(q)\dot{v} = P(q) - F(q,v) + H(q)^\top u, \tag{1}$$

where $M(q) : \mathbb{R}^{3n} \to \mathbb{R}^{3n\times 3n}$ is the mass matrix; $F(q,v) : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ are the internal forces applied to the robot's structure; $P(q) : \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ are the external forces; $H(q) : \mathbb{R}^{3n} \to \mathbb{R}^{3n\times m}$ contains the directions of the forces from the actuators and $u \in \mathbb{R}^m$ is the amplitude of the forces from the actuators. For the robot in Fig. 1, the actuator has dimension $m = 3$.

Denoting $x = [q^\top, v^\top]^\top \in \mathbb{R}^{6n}$, the model (1) can be reformulated in a general form as:

$$\begin{cases} \dot{x} = \begin{bmatrix} v \\ M(q)^{-1}(P(q) - F(q,v) + H(q)^\top u) \end{bmatrix} = G(x,u), \\ y = Cx, \end{cases} \tag{2}$$

where $y \in \mathbb{R}^3$ is the 3D-displacement of the robot's end-effector; and $C \in \mathbb{R}^{3\times 6n}$ is the output matrix for picking out the end-effector from all mesh nodes.

C. Model Order Reduction

Note that model (2) is underactuated, which is not suitable for the synthesis of subsequent control design. Thus, model order reduction will be performed on this model. Since the robot in Fig. 1 is similar to the one in our previous work of [14], the model order reduction process can be achieved by following a similar procedure as presented in [14] and is omitted here. The model reduction results are given in the following for completeness. Specifically, for the system (2), according to [14, Sec. II-A], with the POD method, the state $x \in \mathbb{R}^{6n}$ can be decomposed into two parts: a low-order state $x_r \in \mathbb{R}^r$ (with $r \ll 6n$) and a negligible state $x_{\bar{r}} \in \mathbb{R}^{6n-r}$, such that

$$x = \begin{bmatrix} V_r & V_{\bar{r}} \end{bmatrix} \begin{bmatrix} x_r \\ x_{\bar{r}} \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} x_r \\ x_{\bar{r}} \end{bmatrix} = \begin{bmatrix} U_r & U_{\bar{r}} \end{bmatrix}^\top x \tag{3}$$

with some projectors $V_r \in \mathbb{R}^{6n\times r}$, $V_{\bar{r}} \in \mathbb{R}^{6n\times(6n-r)}$, $U_r \in \mathbb{R}^{6n\times r}$, $U_{\bar{r}} \in \mathbb{R}^{6n\times(6n-r)}$, satisfying $[U_r, U_{\bar{r}}]^\top [V_r, V_{\bar{r}}] = I$. Based on (3), the model (2) can be reformulated as:

$$\begin{cases} \dot{x}_r = U_r^\top G(x_r, x_{\bar{r}}, u), \\ \dot{x}_{\bar{r}} = U_{\bar{r}}^\top G(x_r, x_{\bar{r}}, u), \end{cases} \quad \text{with} \quad y = CV_r x_r + CV_{\bar{r}} x_{\bar{r}}. \tag{4}$$

By neglecting the state $x_{\bar{r}}$, we have $x \approx V_r x_r$, and the model (4) can be approximated by a reduced-order model:

$$\begin{cases} \dot{x}_r = U_r^\top G(x_r, u) = G_r(x_r, u), \\ y = CV_r x_r = C_r x_r. \end{cases} \tag{5}$$

According to [14, Sec. IV-D1], the state x_r in (5) is of dimension $r = 6$ for the robot object of this study.

Remark 2: The FEM technique of Section III-B and the POD technique of Section III-C are used to theoretically deduce the dynamics of the soft robot as the analytical model of (5). This can facilitate the subsequent design of our learning control scheme.

D. Model Linearization

Since we focus on the tracking control of the displacement of the end-effector of the robot, i.e., system output y in (5), we rewrite model (5) into the following form:

$$\dot{y} = C_r \dot{x}_r = C_r G_r(y + x_r - C_r x_r, u) = G_y(y, x_r, u), \tag{6}$$

where $y \in \mathbb{R}^3$, $x_r \in \mathbb{R}^6$, and $u \in \mathbb{R}^3$. Then, consider a stable equilibrium point of robot's operation, i.e., $(y,u) = (0,0)$ when the robot's end-effector points vertically down without external actuation. Through the linearization process around this equilibrium point, model (6) can be derived as:

$$\dot{y} = Ay + Bu + f(x_r), \tag{7}$$

where $A = \frac{\partial G_y}{\partial y}|_{y=0} \in \mathbb{R}^{3\times 3}$, $B = \frac{\partial G_y}{\partial u}|_{u=0} \in \mathbb{R}^{3\times 3}$, and $f(x_r) : \mathbb{R}^6 \to \mathbb{R}^3$ is the linearization error. The matrices A, B can be obtained by performing a linear regression process on the robot around the equilibrium point $(y,u) = (0,0)$, while the function $f(x_r)$ is the system model uncertainty mainly related to the system state x_r, which cannot be precisely known due to the difficulty of modeling soft robot's dynamics.

Remark 3: To facilitate the subsequent control design, we assume the input matrix $B \in \mathbb{R}^{3\times 3}$ in (7) to have full rank. This can be satisfied for the soft trunk robot in Fig. 1, by properly placing the position of the actuator/controller.

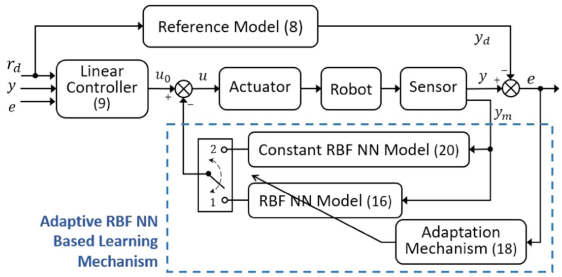


Fig. 2. Block diagram of the adaptive RBF NN-based learning control scheme, involving two operation modes: the adaptive learning control mode (mode 1) and the knowledge-based control mode (mode 2). The control scheme is first operating in the adaptive learning control mode; and then switching to the knowledge-based control mode after the learning control process is completed and the learned knowledge is obtained. y is the displacement of robot's end-effector; y_d is the reference trajectory for y; $e = y - y_d$ is the tracking error; r_d is the reference command in (8); y_m is the output measurement (i.e., the displacement and velocity variables of the robot's end-effector); u_0 is the linear control signal in (9); and u is the total control signal in (17) or (21).

E. Control Objectives

For the soft robot in (7), we design a reference model to generate the desired reference trajectory as follows:

$$\dot{y}_d = A_d y_d + B_d r_d, \tag{8}$$

where $r_d \in \mathbb{R}^3$ is the reference command that can be freely designed, $y_d \in \mathbb{R}^3$ is the generated reference signal, and $A_d \in \mathbb{R}^{3\times3}$, $B_d \in \mathbb{R}^{3\times3}$ are design parameters. This reference model ensures: i) the signal y_d is recurrent, i.e., periodic or almost-periodic; and ii) the parameters A_d, B_d satisfy $A_d = A - BK_y$, $B_d = BK_u$ for some matrices $K_y \in \mathbb{R}^{3\times3}$, $K_u \in \mathbb{R}^{3\times3}$, with A, B given in (7). These conditions are easily satisfied when the matrix B has full rank, which can be guaranteed by our soft robot according to Remark 3.

In this letter, our objective is to drive the displacement of the robot's end-effector y in (7) to accurately track the reference trajectory y_d in (8). For a desired tracking control performance, we need to handle the negative effect of the system uncertainty $f(x_r)$ on the tracking control process. Thus, we will propose an adaptive RBF NN learning-based control scheme aiming to: i) provide stable and accurate control for the end-effector y to track the reference trajectory y_d; and ii) achieve accurate learning/identification for the uncertain dynamics $f(x_r)$ of (7) in the online control process. Then, once the learning control process is completed, we will obtain the learned knowledge to develop a new knowledge-based controller, aiming to provide an efficient and accurate tracking control for the soft robot (7) without needing to repeat any online parameter adaptations.

IV. Adaptive NN-Based Learning Control Design

This section will present the design of the proposed adaptive learning control scheme for the soft robot in Fig. 1. A block diagram illustrating our control scheme is given in Fig. 2.

A. Linear Control Design

We first consider the case that the model (7) has a small linearization error $f(x_r)$ around the equilibrium point $(y, u) = (0, 0)$. Considering the reference model (8), we can design a linear controller as:

$$u_0 = -K(y - y_d) + K_u r_d - K_y y, \tag{9}$$

where y_d, r_d, K_u and K_y are given in (8), and $K \in \mathbb{R}^{3\times3}$ is a design parameter. With this controller, from (7)–(8), the closed-loop error dynamic system (with $e = y - y_d$) can be derived as:

$$\dot{e} = (A_d - BK)e + f(x_r). \tag{10}$$

This implies that when the nonlinear uncertainty $f(x_r)$ of (7) is bounded and small, by selecting a control parameter K of (9) to satisfy $\mathrm{eig}(A_d - BK) < 0$, the closed-loop system (10) can be stable and the tracking error $e = y - y_d$ can be small. However, in real applications, the nonlinear uncertainty $f(x_r)$ of (7) could be large, especially when i) the robot encounters external disturbances, e.g., payload and external interaction, resulting in a major change in the system model (7); ii) the robot's end-effector operates in a region away from the equilibrium point $(y, u) = (0, 0)$; and/or iii) the robot has a complicated underlying structure, complex geometry and nonuniform shape, resulting in a highly nonlinear dynamic model. In these cases, since the controller (9) is not designed to handle the uncertainty $f(x_r)$, it could not guarantee a desired tracking control performance.

B. Adaptive NN Learning Based Control Design

To handle the nonlinear uncertainty $f(x_r)$ of (7), we propose to incorporate an adaptive RBF NN-based learning mechanism into the linear controller (9), so as to generate the adaptive RBF NN-based learning controller as shown in Fig. 2. To this end, from Remark 3, we rewrite the model (7) as:

$$\dot{y} = Ay + B\left(u + B^\top \left(BB^\top\right)^{-1} f\left(x_r\right)\right). \tag{11}$$

For the uncertain function $B^\top(BB^\top)^{-1}f(x_r)$, according to Section II, there exists an ideal constant NN weight $W^* \in \mathbb{R}^{N_n\times3}$ (with N_n denoting the number of NN nodes) such that

$$B^\top \left(BB^\top\right)^{-1} f(x_r) = W^{*\top} S(x_r) + \epsilon, \tag{12}$$

where $S(x_r) \in \mathbb{R}^{N_n}$ is an RBF vector, and $\epsilon \in \mathbb{R}^3$ is an ideal approximation error.

Based on (12), the NN learning process will require the signal x_r to be measurable as NN input. This is difficult in practice, because the robot's whole state x as well as the reduced-order state $x_r = U_r^\top x$ might not be measurable. In view of this, we will take a simple variable-transformation as follows. We set a sufficient number of sensors on the robot in Fig. 1 to obtain a measurement of $y_m \in \mathbb{R}^{\bar{r}}$ (with $\bar{r} \geq 6$). Then, from (3), we have

$$y_m = C_y x \approx C_y V_r x_r = C_m x_r, \tag{13}$$

where $C_y \in \mathbb{R}^{\bar{r}\times6n}$, $C_m \in \mathbb{R}^{\bar{r}\times6}$ are the output matrices for picking out the measurement y_m respectively from the robot's state x and x_r. By appropriately arranging the sensors such that the matrix C_m is of full rank, from (13), we have: $x_r =$

$(C_m^\top C_m)^{-1}C_m^\top y_m$, and the model (11) can be formulated as:

$$\dot{y} = Ay + B\left(u + B^\top\left(BB^\top\right)^{-1} f\left(\left(C_m^\top C_m\right)^{-1} C_m^\top y_m\right)\right)$$
$$= Ay + B\left(u + B^\top\left(BB^\top\right)^{-1} f\left(y_m\right)\right), \quad (14)$$

and the NN approximation of (12) can be rewritten as

$$B^\top\left(BB^\top\right)^{-1} f(y_m) = W^{*\top} S(y_m) + \epsilon. \quad (15)$$

Based on this, we can implement the NN learning of $B^\top(BB^\top)^{-1}f(y_m)$ by using the robot's measurement y_m instead of the robot's state x_r.

Remark 4: For (13), to guarantee C_m to have full column rank, we require the measurement y_m to satisfy $\dim(y_m) \geq \dim(x_r)$. In view of this, considering the soft robot in Fig. 1, whose dominant state x_r has dimension of $r = 6$ in (5), we can set the sensors on the robot to obtain 6-dimensional measurement y_m (e.g., the 3D displacement and velocity of the robot's end-effector). As long as the sensors are properly placed, the matrix C_m of (13) can have full rank.

From (15), we can construct an adaptive RBF NN model to identify/learn the nonlinear uncertainty $f(y_m)$ as follows:

$$B^\top(BB^\top)^{-1}\hat{f}(y_m) = \hat{W}^\top S(y_m), \quad (16)$$

where $\hat{W} \in \mathbb{R}^{N_n \times 3}$ is the estimate of the ideal NN weight W^* in (15) and $\hat{f}(y_m)$ is the estimate of $f(y_m)$ in (14). By incorporating this model with the controller (9), we develop an adaptive NN learning-based controller as:

$$u = -K(y - y_d) + K_u r_d - K_y y - \hat{W}^\top S(y_m), \quad (17)$$

where K is a design parameter satisfying $\text{eig}(A_d - BK) < 0$; y_d, r_d, A_d, K_u, K_y are given in reference model (8); and the NN weight $\hat{W}$ is updated with the following adaptation law:

$$\dot{\hat{W}} = \Gamma S(y_m)(y - y_d)^\top B - \Gamma\sigma\hat{W}, \quad (18)$$

where B is given in (14), $\Gamma = \Gamma^\top > 0$ and $\sigma > 0$ are design parameters, with σ being of a small value.

The learning controller (17)–(18) is designed by extending our previous works in [10], [11], based on a positive-definite Lyapunov function: $V = e^\top e + tr(\tilde{W}^\top\Gamma^{-1}\tilde{W})$ with $e = y - y_d$ and $\tilde{W} = \hat{W} - W^*$. The overall performance of control stability and signals' convergence can be theoretically proved based on the deterministic learning theory [16], and the details are omitted here. Under the control of (17)–(18), from (14) and (8), we derive the error system dynamics as:

$$\dot{e} = (A_d - BK)e + f(x_m) - B\hat{W}^\top S(y_m). \quad (19)$$

It shows that the effect of uncertainty $f(y_m)$ on the control process can be compensated by the NN model $\hat{W}^\top S(y_m)$, thus the tracking error e can be made small. This verifies that, compared to the linear controller (9), the adaptive NN learning controller (17) could provide a better tracking control performance for the soft robot in real applications.

C. Knowledge Based Control Design

With the adaptive learning control system of (8), (14), (17) and (18), once the learning control process is completed, we can obtain the learned knowledge to design a new knowledge-based controller as follows.

According to the analysis in [10], [11], [16], the adaptive learning controller (17)–(18) can achieve an accurate learning/identification for the nonlinear uncertainty $B^\top(BB^\top)^{-1}f(y_m)$ of (14) during the control process. That is, the adaptive NN model $\hat{W}^\top S(y_m)$ can provide a locally-accurate approximation for $B^\top(BB^\top)^{-1}f(y_m)$, and the NN weight $\hat{W}$ can converge to a small neighborhood of the optimal value W^* through the control process. Based on this, the learned knowledge of $B^\top(BB^\top)^{-1}f(y_m)$ can be obtained and stored in a constant RBF NN model $\bar{W}^\top S(y_m)$, i.e.,

$$B^\top(BB^\top)^{-1}f(y_m) \approx \bar{W}^\top S(y_m), \quad (20)$$

where $\bar{W}$ is the convergent value of NN weight $\hat{W}$ in (17), which can be obtained by $\bar{W} = \text{mean}_{t\in[t_1,t_2]}\hat{W}(t)$ with $[t_1, t_2]$ being a time segment after the transient process.

Using the constant RBF NN model (20), following a similar line of the design of (17), we can propose a knowledge-based controller as follows:

$$u = -K(y - y_d) + K_u r_d - K_y y - \bar{W}^\top S(y_m), \quad (21)$$

where K, K_u and K_y are design parameters given in (17). Similar to the case of (17)–(19), the controller (21) can also provide a desired control performance for the robot by using the model $\bar{W}^\top S(y_m)$ to handle the uncertainty $f(y_m)$ in (14). Particularly, different from the adaptive learning controller (17)–(18), the control process of (21) does not need to repeat any online adaptation of NN parameters, thereby reducing the computational complexity and time cost.

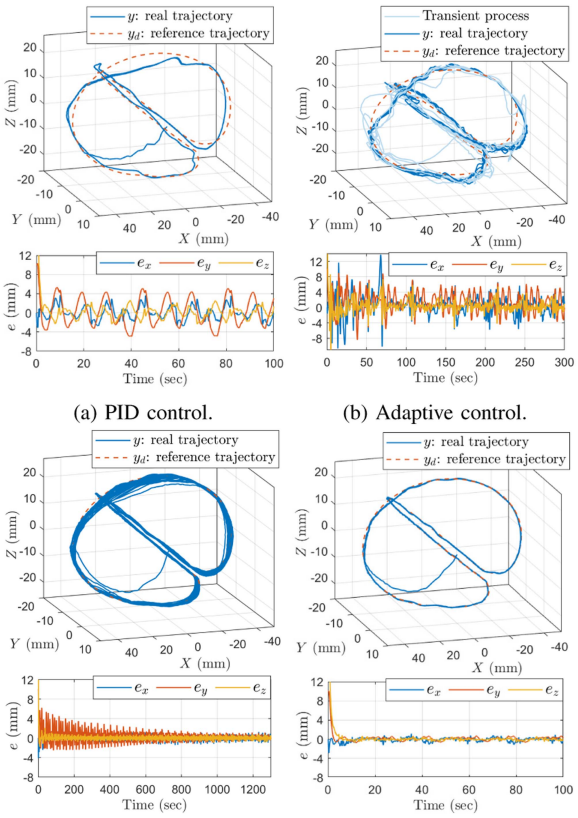
Remark 5: We emphasize that our schemes (including the learning controller of (17)–(18) and the knowledge-based controller of (21)) can be used even when the soft robot is operating in a region away from the equilibrium point $(y, u) = (0, 0)$. This is due to the adaptive RBF NN-based learning mechanism incorporated in the controller. It can online learn the robot's uncertain dynamics $f(x_r)$ in (7) and compensate its effect on the control process in real time, so as to guarantee the desired system stability and tracking accuracy. Such setup can facilitate our scheme to provide desired control performance when the robot moves away from the equilibrium point and its modeling uncertainty (i.e., the uncertain dynamics $f(x_r)$) becomes larger.

V. Experimental Testing on Soft Robot

In this section, we will perform physical experiments with the setup shown in Fig. 1, which includes the soft trunk robot, the actuation system (air tubes) and the sensor system (reflective balls and infrared cameras). The displacement of the robot's end-effector is measured by using OptiTrack motion capture systems [17], and the velocity is obtained with the finite difference approximation technique.

A. Experiment 1: Feasibility Test

The first experiment is to validate the effectiveness and advantages of our approach. We examine the tracking performance of the soft robot under PID control, adaptive control, and adaptive



(a) PID control. (b) Adaptive control.

(c) Adaptive learning control. (d) Knowledge-based control.

Fig. 3. Tracking control performance for the soft robot using different control methods.

TABLE I
STEADY-STATE TRACKING ERROR OF THE SOFT ROBOT UNDER DIFFERENT CONTROL METHODS

Control methods	Max ($\|e\|$)	Mean ($\|e\|$)	RMS ($\|e\|$)
PID control	6.232 mm	3.119 mm	3.360 mm
Adaptive control	7.467 mm	2.931 mm	3.189 mm
Adaptive NN learning control	1.661 mm	0.444 mm	0.499 mm
Knowledge-based control	1.457 mm	0.489 mm	0.536 mm

$\|e\|$ is the Euclidean norm of tracking error $e = y - y_d$; RMS is the Root-mean-square value.

RBF NN learning control and knowledge-based control, respectively. The comparison results are presented in Fig. 3 and in Table I.

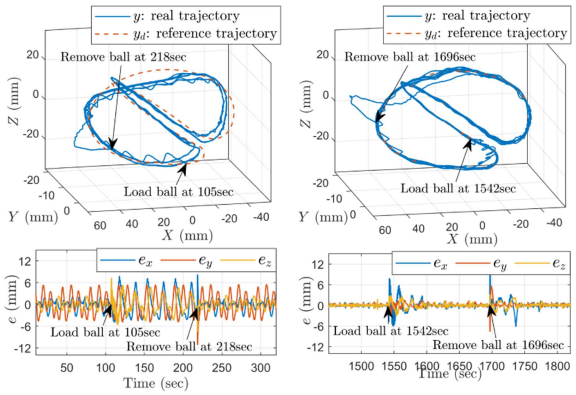
1) Model Linearization and Reference Model Design: For control implementation, we need to derive the soft robot's linearized model in (7). They are obtained by performing a linear-regression process on the robot in Fig. 1 around the equilibrium point $(y, u) = (0, 0)$, i.e., the robot's end-effector points vertically down without external actuation. We have $A = [77, 8.23, -4.47; -0.56, 94.9, -1.14; -11.8, 55.4, 100] \cdot 10^{-3}$, $B = [-71.9, -1.09, 59.1; -13.8, -7.95, -15.3;\ 35.2,\ -74.3, 37.1] \cdot 10^{-2}$. The reference model (8) is given with $A_d = 0$, $B_d = B$, and the reference trajectory y_d is plotted as a red dash line in Fig. 3.

2) PID Control: We first examine the control performance for the soft robot using the PID control. The control gain is set as: $P = [-0.8, -0.7, 0.2; 0, -0.7, -1; 0.8, 0.7, 0.2]$, $I = 0.01P$ and $D = 0$. The tracking performance is shown in Fig. 3(a), in which the robot can be driven to track the reference path, but the tracking error is relatively large (± 6 mm) and presents an oscillatory behavior. It cannot accurately deal with the robot's complex nonlinear uncertain dynamics without parameter tuning, thereby leading to poor tracking performance.

3) Adaptive Control: We further examine the control performance using the model reference adaptive control approach borrowed from [18]. The tracking performance for the soft robot is plotted in Fig. 3(b), showing a limited tracking accuracy level that is similar to the case of PID control (see Table I). This result implies that the adaptive controller without the learning capability cannot effectively adapt to the complex highly-uncertain nonlinear dynamics of the robot in the control process, failing to provide the desired control performance.

4) Adaptive RBF NN Learning Control: Next, we examine the control performance of our proposed adaptive NN learning controller (17)–(18). The RBF NN model $\hat{W}^\top S(y_m)$ is constructed in a regular lattice with the nodes $N_n = 531441$, and the NN input y_m is the real-time displacement and velocity variables of the robot's end-effector. Associated design parameters are given as $K = [-8.29, -8.13, 3.82; -0.12, -8.84, -9.69; 7.01, -9.89, 4.07] \cdot 10^{-1}$, $K_y = [-6.3, -22.1, 3.33; 1.3, -30.9,\ -10.33; 5.39, -26, 3.1] \cdot 10^{-2}$, $K_u = I$, $\Gamma = 0.4$, and $\sigma = 0.00125$. The tracking control performance of the adaptive learning controller is illustrated in Fig. 3(c), where the robot's end-effector y can accurately track the reference trajectory y_d, and the tracking error $e = y - y_d$ decreases over time and converge to a small range (± 1.5 mm). As shown in Table I, the adaptive learning controller can provide a steady-state tracking error with max value 1.661 and mean value 0.444, which is smaller than those of PID control (max value 6.232 and mean value 3.119) and adaptive control (max value 7.467 and mean value 2.931). These results verify that, compared to the traditional PID and adaptive controllers, our adaptive NN learning controller can effectively handle the soft robot's complex dynamics and provide an improved control performance, i.e., higher tracking accuracy.

5) Knowledge-Based Control: Once the learning control process of (17)–(18) is completed, we further examine the knowledge-based controller (21) with the control parameters (K, K_y, K_u) given the same as above. The results are presented in Fig. 3(d) and Table I. The knowledge-based controller can achieve a similar tracking accuracy compared to the adaptive NN learning controller (see Table I), but provide a faster tracking speed (see Fig. 3(c), and (d)). This is because, different from the adaptive NN learning controller, the knowledge-based controller is integrated with the knowledge learned through the control process, which can provide a control action by quickly recalling the embedded knowledge without repeating any online parameter adaptation, thus leading to desirable tracking accuracy and faster convergence rate.



(a) PID control. (b) Adaptive learning control.

Fig. 4. Tracking control performance using different control methods for the soft robot in Fig. 1 when encountering external disturbances during online operation.

B. Experiment 2: Adaptability Test

The second experiment is to show the adaptability of our adaptive learning controller (17)–(18) for the soft robot against external disturbances. During the operation of the soft robot, we attach a blue ball (0.102 kg) on the robot (0.142 kg) for a period of time, which results in a major change in the robot's dynamic model. For this control situation, we examine the PID controller and our adaptive learning controller by using the same control setup as above, and the results are plotted in Fig. 4. Specifically, with the PID controller in Fig. 4(a), after loading the blue ball at about 105 sec, the robot's tracking error will increase and remain stationary until the ball is removed at about 218 sec. This indicates that the PID controller cannot effectively cope with the robot's model change induced by the blue ball. With our adaptive learning controller in Fig. 4(b), after loading the blue ball at about 1542 sec, the robot's tracking error will first increase and then quickly decrease to a level similar as before within about 50 sec. A similar change in the tracking error is observed when removing the ball at about 1696 sec. This illustrates that once the robot encounters external disturbances, our learning controller can quickly adapt and respond to the dynamics change, and maintain the tracking accuracy at a desirable level. Consequently, the results in Fig. 4 verify that our adaptive learning controller can provide a better control performance for the soft robot against external disturbances compared to a traditional PID controller.

C. Experiment 3: Practicality Test

The third experiment is designed to show the practicality of our control scheme when it operates on different trajectories within the workspace. To this end, we first consider four reference trajectories with different shape and location in the workspace, as shown in Fig. 5. To drive the robot's end-effector to track these trajectories, we first implement the adaptive NN learning control process by using the same control setup of Section V-A4, and obtain the learned knowledge/model to construct the knowledge-based controller of (21). The tracking performance of the knowledge-based control is plotted in Fig. 5,

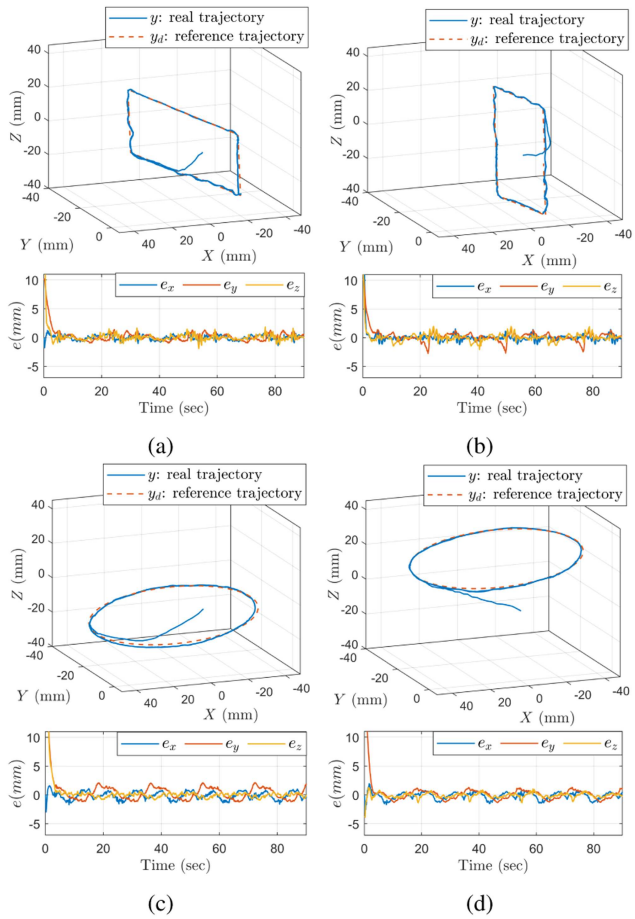
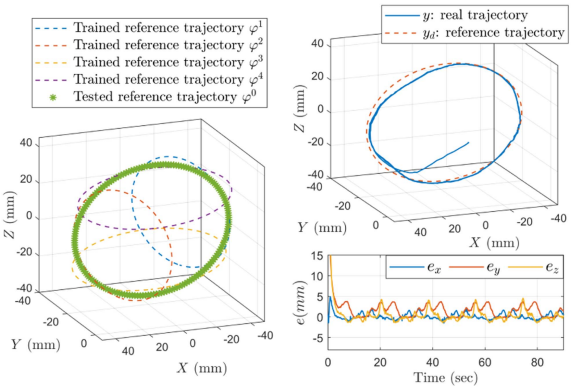


Fig. 5. Tracking control performance of the knowledge-based controller for the soft robot to track different reference trajectories.

showing that the robot can accurately track these different reference trajectories, and the tracking errors converge to a small range (±2 mm). This verifies the applicability of our control scheme on the soft robot operating in the entire workspace.

Next, we will verify the applicability of the learned knowledge (obtained from the adaptive learning control) on the soft robot, by testing it on a new trajectory that has not been trained. We consider five different reference trajectories in Fig. 6(a), including four trajectories φ^i $(i = 1, 2, 3, 4)$ to be trained and one trajectory φ^0 to be tested. We first train the soft robot on the reference trajectories φ^i by following a similar procedure of Section V-A4 and obtain the associated knowledge. By merging these knowledge with the merging mechanism as adopted in [19], we can construct the knowledge-based controller of (21). Then, we test this controller on the soft robot to track the new reference trajectory φ^0, which is different from the trajectories φ^i $(i = 1, 2, 3, 4)$ and has not been trained, and the results are shown in Fig. 6(b). Our control scheme can still drive the robot to track the new trajectory φ^0 while guaranteeing the desired system stability and tracking accuracy. This verifies that using our learning control scheme, the learned knowledge can accurately capture the complex dynamics of the soft robot in the workspace, thus can guarantee the desired practicality of working on the new trajectory.



(a) Reference trajectories. (b) Tracking performance to the trajectory φ^0.

Fig. 6. Tracking control performance of the soft robot using the knowledge-based controller. The knowledge for control design is learned from the reference trajectories φ^i $(i = 1, 2, 3, 4)$ and tested on the new trajectory φ^0.

VI. Conclusion

This letter has developed an adaptive RBF NN based learning control scheme for a soft trunk robot, aiming to drive the end-effector of the robot to track a predefined reference trajectory. Specifically, a low-order approximate model to describe the robot's dynamics has been derived with FEM and POD techniques. Based on this model, an adaptive learning control scheme has been developed with RBF NN technique. It can not only provide stable and accurate tracking control for the soft robot, but also achieve accurate learning for the robot's uncertain dynamics in the online control process. The learned knowledge of robot's dynamics can be stored and represented by a constant RBF NN model. Using this model, a knowledge-based controller has also been proposed, to provide desirable tracking control for the soft robot in an efficient manner. The effectiveness and advantages of the proposed methods have been validated through physical experiments.

In future work, we expect to combine the proposed control methods with the fault diagnosis scheme developed in our previous work of [14], aiming to develop a fault tolerant control scheme for providing safer and more reliable soft robot operations. After this, we plan to extend these approaches from single-segment soft robots to more complex multi-segment ones, for better practical applicability.

References

[1] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Appl. Bionics Biomech.*, vol. 5, no. 3, pp. 99–117, 2008.

[2] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robot.*, vol. 5, no. 2, pp. 149–163, 2018.

[3] R. J. Webster III 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, 2010.

[4] A. Ataka, T. Abrar, F. Putzu, H. Godaba, and K. Althoefer, "Observer-based control of inflatable robot with variable stiffness," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8646–8652.

[5] K.-H. Lee et al., "Nonparametric online learning control for soft continuum robot: An enabling technique for effective endoscopic navigation," *Soft Robot.*, vol. 4, no. 4, pp. 324–337, 2017.

[6] Z. Y. Ding, J. Y. Loo, V. M. Baskaran, S. G. Nurzaman, and C. P. Tan, "Predictive uncertainty estimation using deep learning for soft robot multimodal sensing," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 951–957, Apr. 2021.

[7] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 124–134, Feb. 2019.

[8] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaav1488.

[9] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Learning dynamic models for open loop predictive control of soft robotic manipulators," *Bioinspiration Biomimetics*, vol. 12, no. 6, 2017, Art. no. 66003.

[10] J. Zhang, C. Yuan, C. Wang, W. Zeng, and S.-L. Dai, "Intelligent adaptive learning and control for discrete-time nonlinear uncertain systems in multiple environments," *Neurocomputing*, vol. 462, pp. 31–45, 2021.

[11] J. Zhang, C. Yuan, C. Wang, P. Stegagno, and W. Zeng, "Composite adaptive NN learning and control for discrete-time nonlinear uncertain systems in normal form," *Neurocomputing*, vol. 390, pp. 168–184, 2020.

[12] G. Fang et al., "Vision-based online learning kinematic control for soft robots using local Gaussian process regression," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1194–1201, Apr. 2019.

[13] Z. Q. Tang, H. L. Heung, K. Y. Tong, and Z. Li, "Model-based online learning and adaptive control for a "human-wearable soft robot" integrated system," *Int. J. Robot. Res.*, vol. 40, no. 1, pp. 256–276, 2021.

[14] J. Zhang, X. Chen, P. Stegagno, and C. Yuan, "Nonlinear dynamics modeling and fault detection for a soft trunk robot: An adaptive NN-based approach," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7534–7541, Jul. 2022.

[15] M. Poewell, *The Theory of Radial Basis Function Approximation*. Oxford, U.K.: Clarendon Press, 1992.

[16] C. Wang and D. J. Hill, *Deterministic Learning Theory for Identification, Recognition, and Control*. Boca Raton, FL, USA: CRC Press, 2009.

[17] G. Nagymáté and R. M. Kiss, "Application of optitrack motion capture systems in human movement analysis: A systematic literature review," *Recent Innov. Mechatron.*, vol. 5, no. 1, pp. 1–9, 2018.

[18] E. Lavretsky and K. A. Wise, "Robust adaptive control," in *Robust and Adaptive Control: With Aerospace Applications*. Berlin, Germany: Springer, 2012, pp. 317–353.

[19] T. Chen, C. Wang, and D. J. Hill, "Rapid oscillation fault detection and isolation for distributed systems via deterministic learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1187–1199, Jun. 2014.

[20] A. T. Mathew et al., "Dynamics of suspended cable driven parallel robots using the geometric variable strain approach," in *Proc. IEEE Int. Conf. Soft Robot.*, 2023, pp. 1–7.

[21] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft Robot.*, vol. 6, no. 6, pp. 790–811, 2019.

[22] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1109–1122, Oct. 2014.

[23] C. D. Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 490–513, 2020.

[24] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, "Soft robot control with a learned differentiable model," in *Proc. IEEE 3rd Int. Conf. Soft Robot.*, 2020, pp. 417–423.

[25] F. Cursi, W. Bai, E. M. Yeatman, and P. Kormushev, "Adaptive kinematic model learning for macro-micro surgical manipulator control," in *Proc. IEEE Int. Conf. Adv. Robot. Mechatron.*, 2022, pp. 494–501.

[26] M. Capotondi, G. Turrisi, C. Gaz, V. Modugno, G. Oriolo, and A. De Luca, "An online learning procedure for feedback linearization control without torque measurements," in *Proc. Conf. Robot Learn.*, 2020, pp. 1359–1368.