

Nonlinear Dynamics Modeling and Fault Detection for a Soft Trunk Robot: An Adaptive NN-Based Approach

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

Abstract—This letter presents a radial basis function neural network (RBF NN) based methodology to investigate the dynamics modeling and fault detection (FD) problems for soft robots. Finite element method (FEM) is first used to derive a mathematical model to describe the dynamics of a soft trunk robot. An adaptive dynamics modeling approach is then designed based on this FEM model by incorporating model-reduction and RBF NN techniques. This approach is capable of achieving accurate identification of the soft robot's highly-nonlinear dynamics, with the identified knowledge being obtained and stored in constant RBF NN models. Finally, a model-based FD scheme is proposed with the modeling results, which can achieve efficient FD for the soft robot whenever it encounters an unknown fault. Note that the proposed methods are generic and usable for general soft robots. Validation of these methods is performed through both computer simulation and physical experiments.

Index Terms—Soft robotics, dynamics modeling, fault detection, adaptive dynamics learning, neural networks.

I. INTRODUCTION

SOFT robots are a unique and emergent type of robots. They are made of soft materials, such as silicone and rubber, with no joint in the structure, and their motion is obtained by deformation of the soft components. Soft robots have an important capability of adjusting their shapes and flexibility to adapt to given tasks or unstructured environments, which is a unique advantage over traditional rigid robots in many applications, such as underwater exploration, search and rescue operations, and safe human-robot interaction. This has motivated many researchers to design soft robots in recent years, e.g., soft robotic fish [1] and soft grippers [2].

During the operation of soft robots, fault detection (FD) techniques are crucial to guarantee their safety and reliability [22].

Manuscript received February 22, 2022; accepted June 7, 2022. Date of publication June 17, 2022; date of current version June 28, 2022. This letter was recommended for publication by Associate Editor H. Hauser and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation under Grant CMMI-1929729. (*Corresponding author: Chengzhi Yuan.*)

Jingting Zhang, 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: jingting_zhang@uri.edu; xiaotian_chen@my.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).

Digital Object Identifier 10.1109/LRA.2022.3184034

It is to make an effective indicator that can identify faulty status of the robotic system, so as to facilitate realizing fault tolerant operation for minimizing performance degradation and avoiding dangerous situations [23]. However, as opposed to the substantially growing body of literature on control design (e.g., [3]–[5]), study of FD on soft robots has gained limited success, with only a few research results being reported in [6]–[9], [24]. The study is still in its primitive stage, leaving many challenging issues that have yet to be adequately addressed. For example, researchers in [9] investigated the FD problem of a special soft robot designed with Shape Memory Alloys, which cannot be extended to more general classes of soft robots. [8] studied the fault detection and identification problem of a soft manipulator, which however required the detected faults' information (e.g., faulty type and data) to be available for pre-training, limiting their wider applicability. In [24], a Koopman operator-based scheme has been developed to detect and identify the faults resulting from the change of robot's tip-load, which unfortunately might not be applicable to other types of faults, e.g., robot's air hose obstruction. [6], [7] developed an FD mechanism for locked-motor faults by monitoring the controller's dynamics. These schemes did not investigate the overall dynamics of the studied robot, as such they cannot detect those faults that could be hidden within the robot's uncertain dynamics, e.g., component faults that could not result in deviation of the controller's dynamics. In view of these, it is of particular interest and importance to develop a more practical FD scheme that can be applied to general soft robots and usable for general classes of faults.

To fulfill this objective, one may use the model-based FD method [10], which has a unique capability of providing a deeper insight into the dynamical behaviors of robots, facilitating more efficient and accurate FD. However, developing such a FD method requires the system model of soft robots to be available, which is a very challenging problem. Unlike traditional rigid robots, soft robots have deformable structure, complex geometry and excessive degrees of freedom, thus their physical model is very difficult to deduce. In recent years, some research efforts have been devoted to employing the Finite Element Method (FEM) for modeling and control of soft robots, e.g., [3]–[5]. In these schemes, the FEM technique is used to discretize the robots' geometry into a finite number of small elements, so as to derive a finite-dimensional model to describe the soft robots' behavior. However, these FEM-based schemes only developed an approximate linear dynamic model—describing the

dynamics of soft robots around some equilibrium points—for further control study. Such a model cannot achieve a sufficient modeling accuracy. It thus could not be suitable for designing an accurate and reliable FD scheme.

To accurately model soft robots' nonlinear dynamics, a promising tool is neural network (NN). Some research results have been published by employing NN techniques for the investigation of soft robots, e.g., [11]–[13], [25]. These schemes utilized NN techniques to model some specific feature of soft robots, e.g., turning motion capability [12] and input-to-position mapping relationship [11], [13], so as to facilitate desired control performance. However, these schemes cannot guarantee a sufficiently-accurate modeling performance, which could not be appropriate for the design of FD scheme. This is because, different from the control problem of [11]–[13], [25] where the NN-modeling errors can typically be compensated by the controllers, these errors often have negative impacts on the FD residual signals, which cannot be structured or decoupled from the occurring fault, leading to possible misjudgment of FD [14]. For developing an efficient FD scheme, achieving accurate modeling for the nonlinear dynamics of soft robots with NN techniques is very important, which however still remains as an open problem to date.

In this letter, we aim at achieving accurate identification/modeling of a soft trunk robot's nonlinear dynamics; and proposing a model-based FD scheme for the soft robot. Specifically, FEM is first used to generate a mathematical model for simulating the overall-state dynamics of soft robot. Based on this FEM model, an adaptive dynamics identification approach is proposed with model-reduction and radial basis function neural network (RBF NN) techniques. In particular, model reduction method is employed to derive a low-order model capturing the robot's dominant dynamics for further NN training purpose; and RBF NN technique is used to design an adaptive dynamics identifier. This identification approach is based on the deterministic learning theory [15], which can be theoretically and practically guaranteed to achieve accurate identification for the soft robot's dynamics, and the identified knowledge can be obtained and stored in constant RBF NN models. With these identification results, a model-based FD scheme will be developed to achieve efficient FD for the soft robot whenever an unknown fault occurs. In particular, the design of this FD scheme only needs the data of the robot operating in the normal mode, while does not requires any data of the robot's operating under any pre-known faulty modes. The FD scheme can thus be applicable for general soft robots with general types of faults. These important features distinguish our scheme from many existing FD schemes, e.g., [6]–[9]. The proposed methods are validated through both computer simulation and physical experiments.

The main contributions of this work are: (i) achieving accurate identification/modeling for the soft robot's overall-state nonlinear dynamics; (ii) proposing a model-based FD scheme for soft robot; and (iii) performing both simulational and experimental validations for the proposed methods.

The remainder of this letter is organized as follows. Section II provides some preliminaries. Section III contains the problem statement. The adaptive NN-based identification scheme is

presented in Section IV. The FD scheme is presented in Section V. Section VI concludes the letter.

II. PRELIMINARIES

A. Proper Orthogonal Decomposition (POD)

POD is a well-known model order reduction method [16]. It has been widely used for nonlinear large-scale systems to deal with their high dimensionality.

Consider a large scale dynamical system:

$$\dot{x} = f(x, u), \quad x \in \mathbb{R}^n. \quad (1)$$

According to the projection-based model reduction method of [16], there exist two projectors $V = [V_r, V_{\bar{r}}] \in \mathbb{R}^{n \times n}$ and $U = [U_r, U_{\bar{r}}] \in \mathbb{R}^{n \times n}$, such that the state x can be decomposed into two parts: a low-order state $x_r \in \mathbb{R}^r$ (with $r \ll n$) and a negligible state $x_{\bar{r}} \in \mathbb{R}^{n-r}$, i.e.,

$$x = \begin{bmatrix} V_r & V_{\bar{r}} \end{bmatrix} \begin{bmatrix} x_r \\ x_{\bar{r}} \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} x_r \\ x_{\bar{r}} \end{bmatrix} = \begin{bmatrix} U_r & U_{\bar{r}} \end{bmatrix}^\top x \quad (2)$$

where $V_r \in \mathbb{R}^{n \times r}$, $V_{\bar{r}} \in \mathbb{R}^{n \times (n-r)}$, $U_r \in \mathbb{R}^{n \times r}$, $U_{\bar{r}} \in \mathbb{R}^{n \times (n-r)}$, and $U^\top V = I_n$. The objective of POD is to find the two projectors V_r and U_r for computing the state $x_r = U_r^\top x$, such that $x \approx V_r x_r$. Then, based on the original system (1), a low-order system can be derived as follows:

$$\dot{x}_r = U_r^\top f(V_r x_r, u) = f_r(x_r, u), \quad x_r \in \mathbb{R}^r, \quad r \ll n. \quad (3)$$

According to [16], the projectors V_r and U_r of (2) can be obtained with the snapshots of systems state x in (1). Let S_x be the collection of $s \in \mathbb{N}^+$ snapshots of x , i.e.,

$$S_x = (x(t_1), x(t_2), \dots, x(t_s)) \in \mathbb{R}^{n \times s}. \quad (4)$$

Then, we perform a singular value decomposition on this matrix, i.e., $S_x = \mathcal{V} \Sigma \Omega^\top$, obtaining the left singular matrix $\mathcal{V}$ and the singular value matrix Σ . According to the decay rate of the singular values of Σ , we can select the first r columns of $\mathcal{V}$ to construct a new matrix $\mathcal{V}_r$ such that:

$$S_x = \mathcal{V} \Sigma \Omega^\top = \mathcal{V}_r \Sigma_r \Omega_r^\top + \Delta \quad (5)$$

with Δ representing the model reduction errors. Based on this, the projectors V_r and U_r in (2) are obtained as:

$$V_r = U_r = \mathcal{V}_r. \quad (6)$$

Using these projectors, the reduced-order system (3) can be derived.

B. Radial Basis Function Neural Network (RBF NN)

From [17], RBF NNs can be described by

$$f_{nn}(x) = \sum_{i=1}^{N_n} \hat{w}_i s_i(x) = \hat{W}^\top S(x) \quad (7)$$

where $x \in \Omega_x \subset \mathbb{R}^n$ is the input vector with Ω_x being a compact set, $\hat{W} = [\hat{w}_1, \dots, \hat{w}_{N_n}]^\top \in \mathbb{R}^{N_n}$ is the weight vector, with N_n denoting the NN node number, and $S(x) = [s_1(\|x - \varsigma_1\|), \dots, s_{N_n}(\|x - \varsigma_{N_n}\|)]^\top : \mathbb{R}^n \rightarrow \mathbb{R}^{N_n}$, with $s_i(\cdot)$ being a

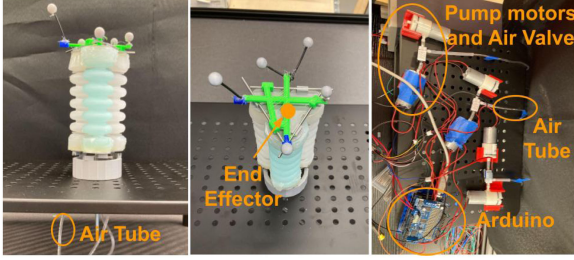


Fig. 1. Soft trunk robot studied in this letter: robot's prototype (left); sensors' setup (middle); and actuators' arrangement (right).

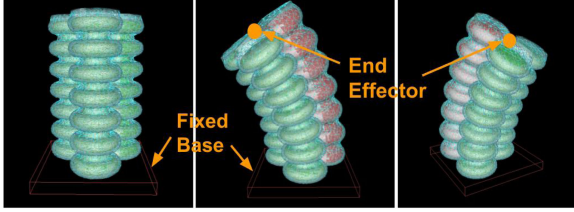


Fig. 2. Soft trunk robot's FEM model built in SOFA. Robot's motion/deformation (middle and right) by pressurizing the air inside the red segment.

radial basis function, and $\varsigma_i \in \mathbb{R}^n$ ($i = 1, 2, \dots, N_n$) being distinct points in state space. In this letter, 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 [17], for any continuous function $f(x) : \Omega_x \rightarrow \mathbb{R}$, and for the NN approximator of (7), where the node number N_n is sufficiently large, there exists an ideal constant weight vector $W^* \in \mathbb{R}^{N_n}$, such that for any $\epsilon^* > 0$,

$$f(x) = W^{*\top} S(x) + \epsilon, \quad \forall x \in \Omega_x \quad (8)$$

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 $\hat{W}$ in (7) that minimizes $|\epsilon|$ for all $x \in \Omega_x$, i.e., $W^* := \operatorname{argmin}_{\hat{W} \in \mathbb{R}^{N_n}} \{\sup_{x \in \Omega_x} |f(x) - \hat{W}^\top S(x)|\}$.

III. PROBLEM STATEMENT

A. Soft Trunk Robot

The studied soft trunk robot is shown in Fig. 1. It is pneumatic actuated, and composed of three identical segments made by high elasticity silicone rubber. The robot can move in the 3D space with the cooperative motion/deformation of these segments, as seen in Fig. 2. Five reflective balls (as sensors) are fixed on the top of the soft robot to locate the robot's end-effector. In terms of robot's each segment, it has maximum length of 108 mm and maximum width of 32 mm. They are able to extend and shrink vertically by pressurizing/depressurizing the air inside the segment. The maximum deformation displacement along vertical direction is about 60 mm. To realize such a motion/deformation, three air tubes are coming out from the bottom of the soft robot to link each segment with two pump

motors and one electrical air valve. One pump motor is for pressurizing the air inside the segment, and the other one is for depressurizing. These motors and valves are all controlled by an Arduino board [18].

B. FEM Model Description

For the deformable robot in Fig. 1, it is difficult to accurately describe its complex geometry and model its continuously deforming structure without using a large state space. We utilize the FEM technique to discretize the robot's structure into a mesh of finite elements, so as to establish an FEM model in Fig. 2. A number of 29626 tetrahedron-type elements are chosen, such that the robot's body can be spatially discretized into $N = 9376$ mesh nodes. Simulation of this model is done using the SOFA framework (an FEM-based simulator) with the SoftRobots Plugin [19].

Define $q \in \mathbb{R}^n$ as the position of each mesh node of the FEM model, and $v \in \mathbb{R}^n$ as the velocity vector of the model, in which the dimension $n = 3 \times N = 28128$ since the variables q, v are considered in the 3D space. Then, by Newton's second law, the nonlinear dynamical model that is used to describe the motion of the robot in Fig. 2 can be given as follows:

$$M(q)\dot{v} = P(q) - F(q, v) + H(q)^\top u, \quad (9)$$

where $\dot{v} \in \mathbb{R}^n$ is the acceleration vector; $M(q) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ is the mass matrix; $F(q, v) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the internal forces applied to the robot's structure; $P(q) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is external forces; $H(q)^\top u$ is the actuators contribution with $H(q) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ containing the direction of actuator's forces and $u \in \mathbb{R}^m$ the amplitude.

Denoting $x = [q, v]^\top \in \mathbb{R}^{2n}$, the system of (9) can be reformulated in the following general form:

$$\dot{x} = f(x) + g(x)u, \quad y = Cx, \quad (10)$$

where $f(x) = [v; M(q)^{-1}P(q) - M(q)^{-1}F(q, v)]$, $g(x) = [0; M(q)^{-1}H(q)^\top]$; $y \in \mathbb{R}^p$ (with $p = 6$) is the measurement (i.e., position and velocity variables) of the robot's end-effector; and $C \in \mathbb{R}^{p \times 2n}$ is the output matrix for picking out the end-effector measurement from all mesh nodes of the FEM model.

Note that the FEM model developed in Fig. 2 is for simulating the dynamical behavior of the real soft robot in Fig. 1. Accurate values of all parameters in the mathematical model (10) are assumed unknown. In this letter, we aim at: (i) developing an adaptive RBF NN-based dynamics identification approach for this model (10), so as to realize accurate identification/modeling for the unknown nonlinear dynamics $f(x)$ and $g(x)$; and (ii) proposing a model-based FD scheme with the modeling results to achieve accurate FD for the system (10) whenever it encounters an unknown fault. A block diagram illustrating the design of our approaches is given in Fig. 3. It is seen that the design of such methods will be performed directly based on the nonlinear dynamical model (10), without needing any linearization process that has been adopted in [3]–[5]. This would facilitate our approach to develop better dynamics-modeling accuracy and FD reliability.

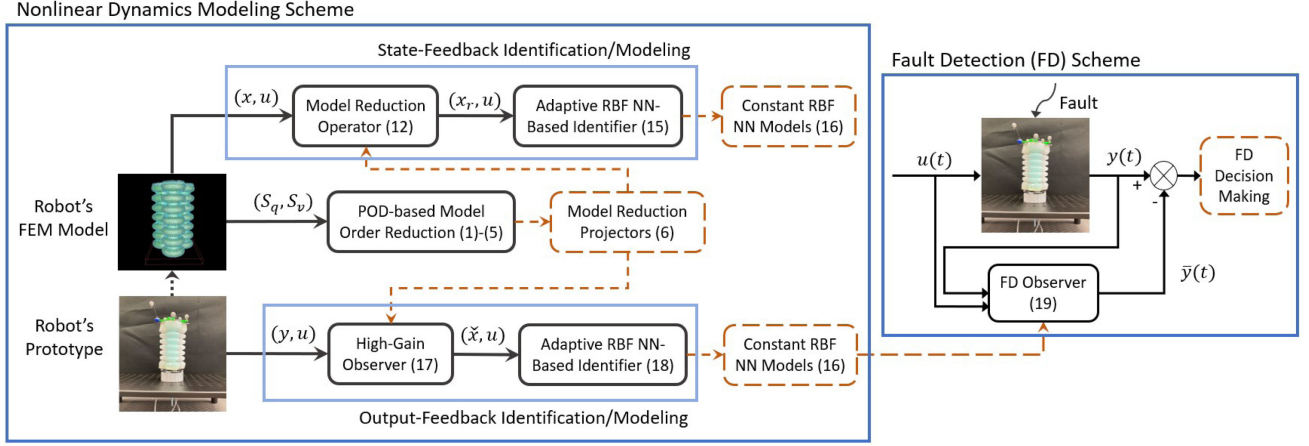


Fig. 3. Block diagram of the nonlinear dynamics modeling scheme and fault detection scheme. $x = [q, v]^T$ is the measurement (i.e., position q and velocity v) of the mesh nodes of robot's FEM model; S_q and S_v are the snapshots of mesh nodes' position q and velocity v , respectively; u is the control input of robot; x_r represents the system reduced-order state; $\hat{x}$ is the estimate of state x_r ; y is the measurement of robot's end-effector; $\hat{y}$ is the estimate of output y .

IV. ADAPTIVE NN DYNAMICS IDENTIFICATION

In this section, we will propose an RBF NN-based identification approach for modeling the soft robot's nonlinear dynamics based on the model (10). The design procedure is shown in Fig. 3. Both the state-feedback and output-feedback cases will be considered.

A. Model Order Reduction

Note that the system (10) is of large dimension, which is not suitable for the subsequent development of NN modeling. Thus, the POD algorithm in Section II-A will be first used for model order reduction purpose. This is an offline process and usually is computationally intensive. However, it will be performed only once to find the desired projectors of (6), and it does not affect real-time execution.

Specifically, by performing the process (4)–(6), we can get the two projectors U_r and V_r . Then, according to the idea in (1)–(3), the system (10) can be rewritten as:

$$\begin{cases} \dot{x}_r = U_r^T f(x_r, x_{\bar{r}}) + U_r^T g(x_r, x_{\bar{r}})u, \\ \dot{x}_{\bar{r}} = U_{\bar{r}}^T f(x_r, x_{\bar{r}}) + U_{\bar{r}}^T g(x_r, x_{\bar{r}})u, \\ y = CV_r x_r + CV_{\bar{r}} x_{\bar{r}}. \end{cases} \quad (11)$$

By neglecting the state $x_{\bar{r}}$, this system can be approximated by a reduced-order model:

$$\dot{x}_r = f_r(x_r) + g_r(x_r)u, \quad y = C_r x_r, \quad (12)$$

where $x_r = U_r^T x \in \mathbb{R}^r$, $f_r(x_r) = U_r^T f(x_r) : \mathbb{R}^r \rightarrow \mathbb{R}^r$, $g_r(x_r) = U_r^T g(x_r) : \mathbb{R}^r \rightarrow \mathbb{R}^{r \times m}$ and $C_r = CV_r \in \mathbb{R}^{p \times r}$.

Note that with the POD process, the reduced-order model of (12) is still not available since the functions $f(\cdot)$ and $g(\cdot)$ of (11) are both unknown. However, the POD process provides the projectors U_r and V_r , which can be used to construct the output matrix $C_r = CV_r$ and to real-time generate the system state $x_r = U_r^T x$, which are necessary for the subsequent development of dynamics modeling.

B. State-Feedback Identification

In this subsection, we consider an ideal case when the system state x_r of (12) are available via $x_r = U_r^T x$. A state-feedback adaptive identification approach will be developed to identify the unknown dynamics $f_r(x_r)$ and $g_r(x_r)$ of (12).

From Section II-B, for the unknown functions $f_r(x_r)$ and $g_r(x_r)$ in (12), we know that there exist two RBF NN models $W_1^{*T} S_1(x_r)$ and $W_2^{*T} S_2(x_r)$ such that:

$$f_r(x_r) = W_1^{*T} S_1(x_r) + \epsilon_1, \quad g_r(x_r) = W_2^{*T} S_2(x_r) + \epsilon_2 \quad (13)$$

where $W_1^* \in \mathbb{R}^{N_n \times r}$, $W_2^* \in \mathbb{R}^{N_n \times r}$, $S_1(x_r) : \mathbb{R}^r \rightarrow \mathbb{R}^{N_n}$, $S_2(x_r) : \mathbb{R}^r \rightarrow \mathbb{R}^{N_n \times m}$. Based on this, we can construct two RBF NN models to respectively identify the unknown functions $f_r(x_r)$ and $g_r(x_r)$ as follows:

$$\hat{f}_r(x_r) = \hat{W}_1^T S_1(x_r), \quad \hat{g}_r(x_r) = \hat{W}_2^T S_2(x_r) \quad (14)$$

where $\hat{W}_1 \in \mathbb{R}^{N_n \times r}$ and $\hat{W}_2 \in \mathbb{R}^{N_n \times r}$ are the estimates of W_1^* and W_2^* , respectively.

Using the RBF NN models (14), we propose to design an adaptive dynamics identifier as follows:

$$\begin{cases} \dot{\hat{x}} = K(\hat{x} - x_r) + \hat{W}_1^T S_1(x_r) + \hat{W}_2^T S_2(x_r)u, \\ \dot{\hat{W}}_1 = -\Gamma_1 S_1(x_r)(\hat{x} - x_r)^T P - \Gamma_1 \sigma_1 \hat{W}_1, \\ \dot{\hat{W}}_2 = -\Gamma_2 S_2(x_r)u(\hat{x} - x_r)^T P - \Gamma_2 \sigma_2 \hat{W}_2, \end{cases} \quad (15)$$

where x_r is the state of reduced-order system (12); P , K , Γ_i and σ_i (with $i = 1, 2$) are design parameters satisfying: $P = P^T > 0$, $K^T P + PK < 0$, $\Gamma_i = \Gamma_i^T > 0$ and $\sigma_i > 0$ being a small number.

The design of identifier (15) is based on a positive-definite Lyapunov function $V = \tilde{x}^T P \tilde{x} + \text{tr}(\tilde{W}_1^T \Gamma_1 \tilde{W}_1) + \text{tr}(\tilde{W}_2^T \Gamma_2 \tilde{W}_2)$ with $\tilde{x} = \hat{x} - x_r$ and $\tilde{W}_i = \hat{W}_i - W_i^*$ ($i = 1, 2$). The overall system stability and signals' convergence can be guaranteed based on the deterministic learning theory [15], which for completeness of presentation is summarized as follows. Using the identifier (15) on the system (12), by setting the

NN input trajectory x_r to be recurrent through the NN training process, we can have: (i) all signals in the closed-loop system remain bounded; (ii) the estimation error $|\hat{x} - x_r|$ will converge to a small neighborhood around the origin; and the NN weight $\hat{W}_i$ will converge to a small neighborhood of its optimal value W_i^* ; and (iii) a locally-accurate approximation of the unknown functions $f_r(x_r)$ and $g_r(x_r)$ can be achieved by $\hat{W}_1^\top S_1(x_r)$, $\hat{W}_2^\top S_2(x_r)$, respectively. Proofs are omitted in this letter, and more details can be seen in [15, Th. 3.1].

Once the offline NN training process is completed, we can further obtain the modeling result as follows. According to [15], owing to the convergence of NN weights $\hat{W}_i$ to optimal values W_i^* , we can obtain their corresponding constant weights $\bar{W}_i := \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \hat{W}_i(\tau) d\tau$ with $[t_1, t_2]$ representing a time segment after the transient process. Two constant RBF NN models $\bar{W}_1^\top S_1(x_r)$ and $\bar{W}_2^\top S_2(x_r)$ can be constructed, which are able to locally accurately approximate the unknown functions $f_r(x_r)$ and $g_r(x_r)$ of (12), i.e.,

$$f_r(x_r) \approx \bar{W}_1^\top S_1(x_r); \quad g_r(x_r) \approx \bar{W}_2^\top S_2(x_r). \quad (16)$$

These models will be used in the next sections for developing a model-based FD scheme.

C. Output-Feedback Identification

The above identification approach is based on the system state $x_r = U_r^\top x$ of (12), whose derivation requires the measurement of the whole state x of (10). This could be difficult in practice due to the large number of variables to measure. In this subsection, we consider a more practical case when only soft robot's inputs u and outputs y are measurable. An output-feedback adaptive identification approach will be developed.

Specifically, by using the measured robot's output $y \in \mathbb{R}^p$, a high-gain observer is first developed as follows:

$$\dot{\tilde{x}} = K_h(\tilde{y} - y), \quad \tilde{y} = C_r \tilde{x}, \quad (17)$$

where $\tilde{x}$ is the estimate of system state x_r in (12), $\tilde{y}$ is the estimate of system output y in (12), C_r is from (12), and K_h is a design parameter satisfying $\text{eig}(K_h C_r) < 0$. Particularly, according to [20], by appropriately selecting a high gain K_h , the signal $\tilde{x}$ can estimate the system states x_r of (12) with arbitrarily small estimation errors.

Using the state $\tilde{x}$ of (17), the identifier is designed as:

$$\begin{cases} \dot{\hat{x}} = K(\hat{x} - \tilde{x}) + \hat{W}_1^\top S_1(\tilde{x}) + \hat{W}_2^\top S_2(\tilde{x})u, \\ \dot{\hat{W}}_1 = -\Gamma_1 S_1(\tilde{x})(\hat{x} - \tilde{x})^\top P - \Gamma_1 \sigma_1 \hat{W}_1, \\ \dot{\hat{W}}_2 = -\Gamma_2 S_2(\tilde{x})u(\hat{x} - \tilde{x})^\top P - \Gamma_2 \sigma_2 \hat{W}_2, \end{cases} \quad (18)$$

where the design parameters P , K , Γ_i and σ_i (with $i = 1, 2$) are selected similar to (15).

From (17) and (18), and according to [15, Chap. 7], it is seen that, as long as the estimation error $\tilde{x} - x_r$ is sufficiently small, the identifier (18) can provide accurate identification for unknown dynamics $f_r(\cdot)$ and $g_r(\cdot)$ of system (12). Associated results of system stability and signals' convergence are similar to those of Section IV-B. Consequently, using the output-feedback identifier (17) and (18), we can also obtain constant NN models

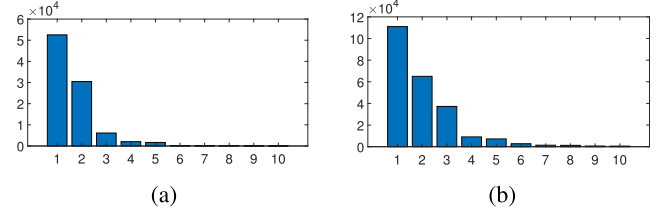


Fig. 4. First 10 singular values of: (a) robot's position snapshots S_q ; and (b) robot's velocity snapshots S_v .

$\bar{W}_1^\top S_1(\cdot)$ and $\bar{W}_2^\top S_2(\cdot)$ to represent the identified knowledge of dynamics $f_r(\cdot)$ and $g_r(\cdot)$, respectively, as seen in (16).

Remark 1: The high-gain observer (17) can achieve accurate estimation for system state x_r in (12) when the output matrix C_r is full column rank, i.e., $\text{rank}(C_r) = r$. This is fulfilled by appropriately arranging the number and position of sensors for our studied robot.

D. Application to Soft Robot

1) *Model Order Reduction:* Model reduction of the soft trunk robot requires measurement of the overall system state, which could be very difficult in practice. Thus, this process is performed on the FEM model of Fig. 2 using the SOFA framework. The obtained results will be valid for both simulated FEM model and the physical robot. Specifically, for our soft robot, we use structure preserving model order reduction for system state $x = [q, v]^\top$ of (10), such that the reduced-order state can keep its initial structure, i.e., $x_r = [x_{r_q}, x_{r_v}]^\top$. The model reduction process will be performed for soft robot's position q and velocity v , respectively. According to the POD algorithm in Section II-A, we store 1200 snapshots of q and v , and construct the snapshot matrices S_q and S_v in the form of (4), both of which have dimension 28128×1200 . Their corresponding singular values are shown in Fig. 4, which decrease very fast. We choose the first 3 values, and obtain the projectors V_{r_q} , V_{r_v} , U_{r_q} , U_{r_v} all with dimension 28128×3 . They guarantee that $x_{r_q} = U_{r_q}^\top q$, $q \approx V_{r_q} x_{r_q}$ and $x_{r_v} = U_{r_v}^\top v$, $v \approx V_{r_v} x_{r_v}$. With these projectors, the large-scale model (10) can be transformed into a low-order model (12), with system dimension being decreased $2n = 2 \times 28128 = 56256 \rightarrow r = 2 \times 3 = 6$.

2) *Simulation Testing:* We use the FEM model of Fig. 2 in SOFA framework to simulate the dynamical behavior of the soft trunk robot, so as to examine the performance of the proposed identification approach, including the state-feedback and output-feedback cases. We first consider the state-feedback case. For the identifier (15), the RBF NNs are constructed in a regular lattice, with the centers evenly spaced on $[-1, 1]$, the nodes $N_n = 531441$, and the widths $\eta_i = 0.25$. Associated parameters are designed as: $K = -1.5$, $P = 1$, $\Gamma_i = 1.5$, and $\sigma_i = 0.001$ ($i = 1, 2$). The simulation results are illustrated in Fig. 5. It shows that the state $\hat{x}$ of the identifier (15) can provide an accurate approximation of the state dynamics x_r .

We further consider the output-feedback identification approach of (17) and (18). The system setup keeps the same as that

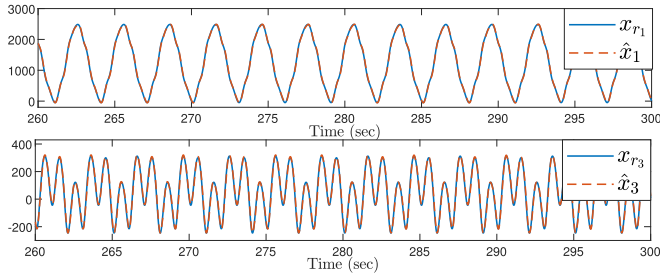


Fig. 5. Simulation testing: approximation performance of state-feedback identifier (15) for system (12): state dynamics $\hat{x} \rightarrow x_r$.

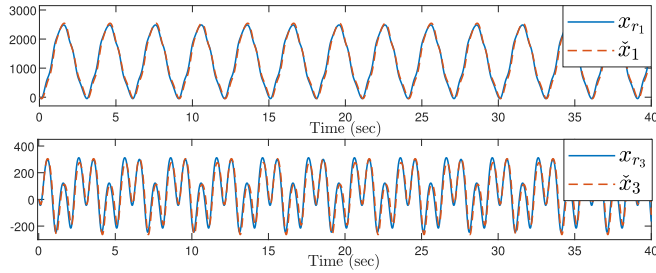


Fig. 6. Simulation testing: approximation performance of output-feedback high-gain observer (17) for system (12): state dynamics $\hat{x} \rightarrow x_r$.

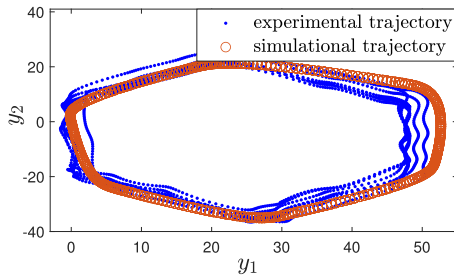


Fig. 7. Comparison of robot's dynamical behavior between FEM model (simulation) and robot's prototype (experiment): output trajectory (y_1, y_2).

of state-feedback case except the parameter $K_h = -20C_r^{-1}$. The performance of high-gain observer (17) is illustrated in Fig. 6, showing that the state $\hat{x}$ can accurately estimate the system state x_r . Using these state estimate $\hat{x}$, accurate identification can be achieved, and associated performances are similar to those in Fig. 5, which are omitted here due to the limited space.

3) *Experimental Testing*: We further consider to use the soft robot's physical prototype in Fig. 1, to test the performance of proposed identification approach. Since the robot's state x is not measurable, the state-feedback identification approach is not usable. We only consider the output-feedback case. We first validate the constructed FEM model by comparing the dynamical behavior between the simulated FEM model of Fig. 2 and the physical prototype of Fig. 1, as seen in Fig. 7, which are close. This guarantees that the model-reduction results of Section IV-D1 are valid on the robot's prototype. Considering the high-gain observer (17) and the identifier (18), associated system setup and parameters are the same as those in the case

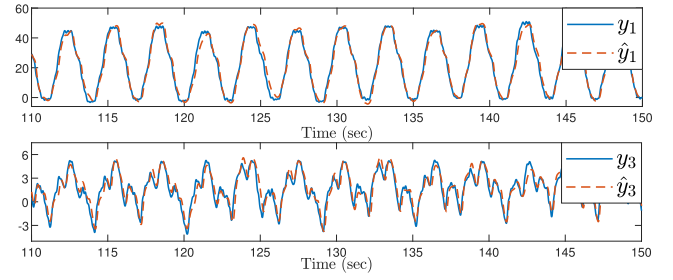


Fig. 8. Experiment testing: approximation performance of output-feedback identifier (18) for soft robot: output dynamics $\hat{y} \rightarrow y$ with $\hat{y} = C_r \hat{x}$.

of simulated FEM model, as mentioned above. The modeling performance is illustrated in Fig. 8. It shows that the robot's real output y can be accurately estimated by the output $\hat{y} = C_r \hat{x}$ of the identifier (18). These justify that our approach can achieve accurate identification for the dynamics of actual robot in Fig. 1.

V. FAULT DETECTION SCHEME

In this section, a model-based FD scheme will be proposed with the modeling result of Section IV, to realize accurate FD for the system (10) whenever an unknown fault occurs. For implementation convenience, this FD scheme will be designed only using the output measurement of system (10).

A. FD Observer Design

From Section IV, we know that two constant RBF NN models $\bar{W}_i^T S_i(\cdot)$ ($i = 1, 2$) can be obtained through offline NN training process. Using these models, we propose to design an output-feedback FD observer as follows:

$$\dot{\hat{x}} = K_y(\bar{y} - y) + \bar{W}_1^T S_1(\hat{x}) + \bar{W}_2^T S_2(\hat{x})u, \quad \bar{y} = C_r \hat{x}, \quad (19)$$

where K_y is a design gain to guarantee that the observer (19) can be stable, which satisfies $\text{eig}(K_y C_r) < 0$.

Consider the observer (19) and the monitored system (12). Note that the constant models $\bar{W}_1^T S_1(\cdot)$ and $\bar{W}_2^T S_2(\cdot)$ in (19) can accurately approximate the unknown functions $f_r(\cdot)$ and $g_r(\cdot)$ in (12), respectively, as seen in (16). Using the observer (19) to monitor the system (12), the state $\hat{x}$ can accurately estimate the system state x_r , and the output $\bar{y}$ can also accurately estimate the system output y , i.e., the error $|\bar{y} - y|$ can be very small. Such an accurate estimation can be guaranteed only when the monitored system (12) is operating in normal mode, i.e., no change occurs in system's dynamics. Thus, the observer (19) can be used to monitor the fault occurrence in system (12) for our FD scheme design.

B. FD Decision Making

The design of our FD scheme is similar to our previous work [21]. Associated implementation procedure is shown in Fig. 3, and is clarified as follows. For the system (10), using the measured system output y and input u , the FD observer (19) can be operated to real-time generate the output $\bar{y}$. Then, comparing the signals $\bar{y}$ of (19) and y of (10), we can obtain

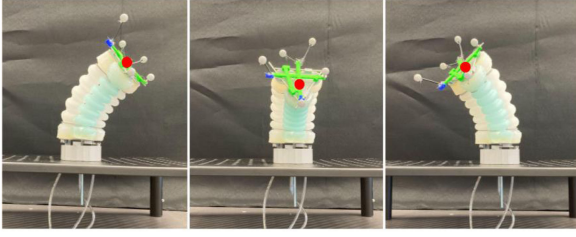


Fig. 9. Soft robot operates under normal mode. The red point is the location of end-effector.

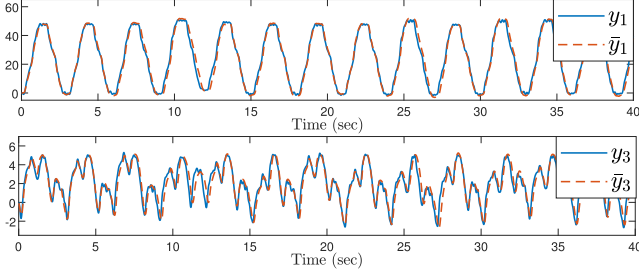


Fig. 10. Approximation performance of FD observer (19) for soft robot: output dynamics $\bar{y} \rightarrow y$.

the FD signal as $\|\bar{y} - y\|_1$, where $\|\cdot\|_1$ represents L_1 norm defined as $\|x(t)\|_1 = \frac{1}{T} \int_{t-T}^t |x(\tau)| d\tau$ with $T > 0$ being a design parameter. With this FD signal, we have: when no fault occurs in the studied robot, $\|\bar{y} - y\|_1$ can be very small, and remain smaller than a given constant threshold, denoted as $\bar{\rho}$. If there exists a time instant t_d , such that the FD signal $\|\bar{y} - y\|_1$ becomes larger than the threshold $\bar{\rho}$, i.e., $\|\bar{y}(t_d) - y(t_d)\|_1 > \bar{\rho}$, it indicates that a certain fault must occur in the robot. Thus, the occurrence of fault can be detected at time t_d . The idea is formalized as follows:

FD decision making: Compare the FD signal $\|\bar{y} - y\|_1$ with the FD threshold $\bar{\rho}$. If there exists a finite time t_d , such that $\|\bar{y}(t_d) - y(t_d)\|_1 > \bar{\rho}$ holds. Then, the occurrence of a fault is deduced at time t_d .

Remark 2: The parameter $\bar{\rho}$ can be determined by trial-and-errors in simulation or experiment study, as will be presented below.

C. Application to Soft Robot

1) Validation of FD Observer: For the FD observer (19), the constant NN models $\bar{W}_i^T S_i$ ($i = 1, 2$) is constructed with constant weights $\bar{W}_i = \frac{1}{100} \int_{1100}^{1200} \hat{W}_i(t) dt$, which are calculated based on the NN weights' convergence. The parameters are set as $K_y = -2C_r^{-1}$. To examine the estimation performance of this observer, the prototype of soft robot in Fig. 1 is used. We enable the soft robot to be operating in normal mode (i.e., no fault occurs) in Fig. 9. Using the FD observer (19) on such a robot, as seen in Fig. 10, accurate estimation for the robot's output y can be achieved by the FD observer's output $\bar{y}$. This validates the effectiveness of our designed FD observer.

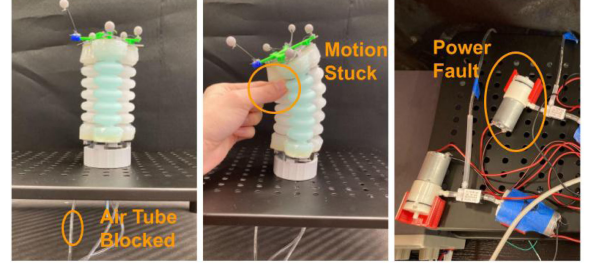


Fig. 11. Soft robot operates under: faulty mode 1 (left); faulty mode 2 (middle); and faulty mode 3 (right).

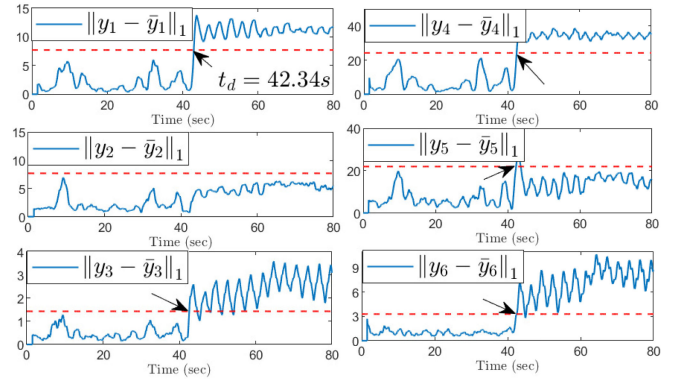


Fig. 12. Fault 1 occurs at time $t = 41.25$ sec, and is detected at time $t_d = 42.34$ sec. Detection time is $t_d - t_0 = 1.09$ sec.

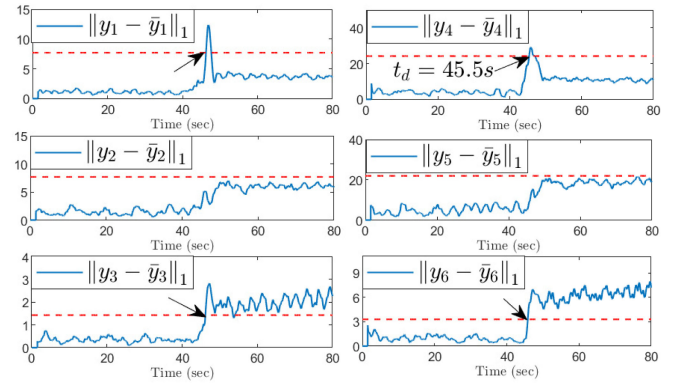


Fig. 13. Fault 2 occurs at time $t = 42.61$ sec, and is detected at time $t_d = 45.5$ sec. Detection time is $t_d - t_0 = 2.89$ sec.

2) FD Performance: FD testing is also based on the prototype of soft robot. As shown in Fig. 11, three different kinds of faults will be considered: (i) faulty mode 1: one of soft robot's air tubes is blocked; (ii) faulty mode 2: the motion/deformation of robot's one segment is stuck due to some external force; and (iii) faulty mode 3: pump motors' power source encounters a fault, i.e., the power voltage raises by 50%. Considering these faults, the FD process will be performed with FD observer (19) and threshold $\bar{\rho} = [7.7, 7.7, 1.43, 24.2, 22, 3.3]$, where the parameter of L_1 norm is $T = 1.5$ sec. Associated FD performances can be seen in Figs. 12–14. In Fig. 12, after fault 1 occurs at time $t = 41.25$ sec, some of the FD signals $\|y_i - \bar{y}_i\|_1$ (with

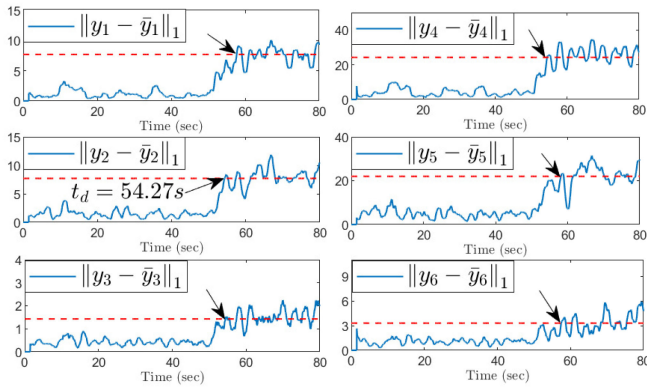


Fig. 14. Fault 3 occurs at time $t = 50.55$ sec, and is detected at time $t_d = 54.27$ sec. Detection time is $t_d - t_0 = 3.72$ sec.

$i = 1, 3, 4, 6$) increase and become larger than their associated threshold; as a result, fault 1 can be detected at time $t_d = 42.34$ sec, and the detection time is $t_d - t_0 = 1.09$ sec. Similar results for the case of fault 2 and fault 3 can also be seen in Figs. 13 and 14. These results justify the effectiveness of our FD scheme on detecting different types of faults occurring in the soft robot.

VI. CONCLUSION

This letter has investigated the dynamics-identification and FD problems of soft robots. Specifically, FEM has been used to derive a mathematical model for simulating the overall-state dynamics of soft robot. Based on the FEM model, an adaptive dynamics identification approach has been proposed by using POD-based model-reduction and RBF NN techniques. This approach can accurately identify the dominant state dynamics of the soft robot, and the identified knowledge can be obtained and stored in constant NN models. Model-based FD scheme has been proposed with these modeling results to achieve efficient FD for the soft robot whenever an unknown fault occurs. The proposed methods have been validated with both simulation testing and experiment testing.

In future work, we expect to extend our proposed dynamics-modeling approach to design a reference-tracking controller for soft robots. After this, we will try to combine this control method with the FD scheme proposed in this letter, to further develop a fault tolerant control scheme for providing more reliable soft robot operations.

REFERENCES

- [1] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus, "Exploration of underwater life with an acoustically controlled soft robotic fish," *Sci. Robot.*, vol. 3, no. 16, 2018, Art. no. eaar3449.
- [2] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *Int. J. Robot. Res.*, vol. 35, no. 1–3, pp. 161–185, 2016.
- [3] K. Wu and G. Zheng, "FEM-based gain-scheduling control of a soft trunk robot," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3081–3088, Apr. 2021.
- [4] M. Thieffry, A. Kruszewski, T.-M. Guerra, and C. Duriez, "LPV framework for non-linear dynamic control of soft robots using finite element model," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7312–7318, 2020.
- [5] M. Thieffry, A. Kruszewski, C. Duriez, and T.-M. Guerra, "Control design for soft robots based on reduced-order model," *IEEE Robot. Automat. Lett.*, vol. 4, no. 1, pp. 25–32, Jan. 2019.
- [6] H. Gu, H. Wang, F. Xu, Z. Liu, and W. Chen, "Active fault detection of soft manipulator in visual servoing," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9778–9788, Oct. 2021.
- [7] H. Gu, H. Wang, and W. Chen, "Toward state-unsaturation guaranteed fault detection method in visual servoing of soft robot manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., IROS*, 2021, pp. 3942–3947.
- [8] H. Gu, H. Hu, H. Wang, and W. Chen, "Soft manipulator fault detection and identification using ANC-based LSTM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., IROS*, 2021, pp. 1702–1707.
- [9] T. S. Le, H. Schlegel, W.-G. Drossel, and A. Hirsch, "Fault detection and fault-tolerant control when using SMA actuators in soft robotics," in *Solid State Phenomena*, vol. 260, Zurich, Switzerland: Trans. Tech. Publ., 2017, pp. 92–98.
- [10] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.
- [11] G. Zheng, Y. Zhou, and M. Ju, "Robust control of a silicone soft robot using neural networks," *ISA Trans.*, vol. 100, pp. 38–45, 2020.
- [12] W. Pang, J. Wang, and Y. Fei, "The structure, design, and closed-loop motion control of a differential drive soft robot," *Soft Robot.*, vol. 5, no. 1, pp. 71–80, 2018.
- [13] 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., RoboSoft*, 2020, pp. 417–423.
- [14] J. Zhang, C. Yuan, P. Stegagno, H. He, and C. Wang, "Small fault detection of discrete-time nonlinear uncertain systems," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 750–764, Feb. 2021.
- [15] C. Wang and D. J. Hill, *Deterministic Learning Theory for Identification, Recognition, and Control*. Boca Raton, FL, USA: CRC, 2009.
- [16] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*. University City, PA, USA: Soc. Ind. App. Math., 2017.
- [17] M. Poewell, *The Theory of Radial Basis Function Approximation*. Oxford: Clarendon Press, 1992.
- [18] J.-D. Warren, J. Adams, and H. Molle, "Arduino for robotics," in *Arduino Robotics*. Cham, Switzerland: Springer, 2011, pp. 51–82.
- [19] E. Coevoet *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Adv. Robot.*, vol. 31, no. 22, pp. 1208–1224, 2017.
- [20] J.-P. Gauthier, H. Hammouri, and S. Othman, "A simple observer for nonlinear systems applications to bioreactors," *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 875–880, Jun. 1992.
- [21] J. Zhang, C. Yuan, W. Zeng, P. Stegagno, and C. Wang, "Fault detection of a class of nonlinear uncertain parabolic PDE systems," *IEEE Control Syst. Lett.*, vol. 5, no. 4, pp. 1459–1464, Oct. 2021.
- [22] A. Abid, M. T. Khan, and J. Iqbal, "A review on fault detection and diagnosis techniques: Basics and beyond," *Artif. Intell. Rev.*, vol. 54, no. 5, pp. 3639–3664, 2021.
- [23] Y.-J. Park, S.-K. S. Fan, and C.-Y. Hsu, "A review on fault detection and process diagnostics in industrial processes," *Processes*, vol. 8, no. 9, 2020, Art. no. 1123.
- [24] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Koopman-based control of a soft continuum manipulator under variable loading conditions," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6852–6859, Oct. 2021.
- [25] J. Y. Loo, Z. Y. Ding, V. M. Baskaran, S. G. Nurzaman, and C. P. Tan, "Robust multimodal indirect sensing for soft robots via neural network-aided filter-based estimation," *Soft Robot.*, vol. 9, no. 3, pp. 591–612, 2022.