Data-Driven Methods Applied to Soft Robot Modeling and Control: A Review

Zixi Chen, *Graduate Student Member, IEEE*, Federico Renda, *Member, IEEE*, Alexia Le Gall, *Student Member, IEEE*, Lorenzo Mocellin, Matteo Bernabei, Théo Dangel, Gastone Ciuti, *Senior Member, IEEE*, Matteo Cianchetti, *Member, IEEE*, and Cesare Stefanini, *Member, IEEE*

***Abstract*— Soft robots show compliance and have infinite degrees of freedom. Thanks to these properties, such robots can be leveraged for surgery, rehabilitation, biomimetics, unstructured environment exploring, and industrial grippers. In this case, they attract scholars from a variety of areas. However, nonlinearity and hysteresis effects also bring a burden to robot modeling. Moreover, following their flexibility and adaptation, soft robot control is more challenging than rigid robot control. In order to model and control soft robots, a large number of data-driven methods are utilized in pairs or separately. This review first briefly introduces two foundations for data-driven approaches, which are physical models and the Jacobian matrix, then summarizes three kinds of data-driven approaches, which are statistical method, neural network, and reinforcement learning. This review compares the modeling and controller features, e.g., model dynamics, data requirement, and target task, within and among these categories. Finally, we summarize the features of each method. A discussion about the advantages and limitations of the existing modeling and control approaches is presented, and we forecast the future of data-driven approaches in soft robots. A website (https://sites.google.com/view/23zcb) is built for this review and will be updated frequently.**

***Note to Practitioners*—This work is motivated by the need for a review introducing soft robot modeling and control methods in parallel. Modeling and control play significant roles in robot research, and they are challenging especially for soft robots. The nonlinear and complex deformation of such robots necessitates specific modeling and control approaches. We introduce the state-of-the-art data-driven methods and survey three approaches widely utilized. This review also compares the performance of these methods, considering some important features like data amount requirement, control frequency, and target task. The features of each approach are summarized, and we discuss the possible future of this area.**

***Index Terms*— Soft robot, data-driven method, physical model, Jacobian matrix, statistical model, neural network, reinforcement learning.**

I. Introduction

SOFT robots have been developed for a large number of applications. Owing to their infinite degrees of freedom (DOFs) and flexibility, soft robots are leveraged in robot assistant surgery, especially minimally invasive surgery [1]. Compared with their rigid counterparts, soft robots are relatively safe due to their softness and have significant advantages in assisting elderly and disabled people with daily tasks [2] and cooperating with humans [3]. Moreover, they are used as hand recovery devices [4] and wearable devices [5] for various medical purposes like rehabilitation and human motion monitoring. Animals are composed of soft tissues, and researchers in the bioinspired area apply soft materials like silicone to build soft robots and mimic the behaviors of living beings, such as octopus [6], elephant [7], and earthworm [8]. These robots produce specific motions and manipulations by imitating the structures and behaviors of soft animals. With the help of such soft biomimetic robots, some exploring tasks in various environments like underwater [9] and walls [10] can be achieved. Soft robot hands are adaptative to objects and commonly applied for grasping tasks involving fragile [11] and diverse [12] objects. To summarize, unique properties like infinite DOFs, compliance, and safety for humans lead to the high potential of soft robots. Hence, soft robot study is a research area highly attractive to robotics scholars.

Followed by the aforementioned advantages and applications, the most considerable drawback of soft robots is the challenge of modeling and control. Deformable materials lead to the nonlinear and delayed responses of soft robots. Moreover, infinite DOFs make it complicated to build accurate models for soft robots. The physical properties of soft robots will also vary due to the aging of soft materials. Owing to the above characteristics, the modeling of soft robots is more

Manuscript received 19 February 2024; accepted 11 March 2024. Date of publication 20 March 2024; date of current version 5 February 2025. This article was recommended for publication by Associate Editor H. Lu and Editor L. Zhang upon evaluation of the reviewers' comments. This work was supported in part by European Union (EU) by the Next Generation EU Project "Ecosistema dell'Innovazione" Tuscany Health Ecosystem [Tuscany Health Ecosystem (THE), Piano Nazionale di Ripresa e Resilienza (PNRR), Spoke 4: Spoke 9: Robotics and Automation for Health] under Grant ECS00000017; in part by U.S. Office of Naval Research Global under Grant N62909-21-1-2033; and in part by Khalifa University under Grant RC1-2018-KUCARS. *(Corresponding author: Zixi Chen.)*

Zixi Chen, Alexia Le Gall, Lorenzo Mocellin, Matteo Bernabei, Théo Dangel, Gastone Ciuti, Matteo Cianchetti, and Cesare Stefanini are with The BioRobotics Institute and the Department of Excellence in Robotics and AI, Scuola Superiore Sant'Anna, 56127 Pisa, Italy (e-mail: zixi.chen@santannapisa.it; alexiamarie.legall@santannapisa.it; lorenzo.mocellin@santannapisa.it; matteo.bernabei@santannapisa.it; theo.dangel@santannapisa.it; gastone.ciuti@santannapisa.it; matteo.cianchetti@santannapisa.it; cesare.stefanini@santannapisa.it).

Federico Renda is with the Khalifa University Center for Autonomous Robotic Systems, Khalifa University, Abu Dhabi, United Arab Emirates (e-mail: federico.renda@ku.ac.ae).

Digital Object Identifier 10.1109/TASE.2024.3377291

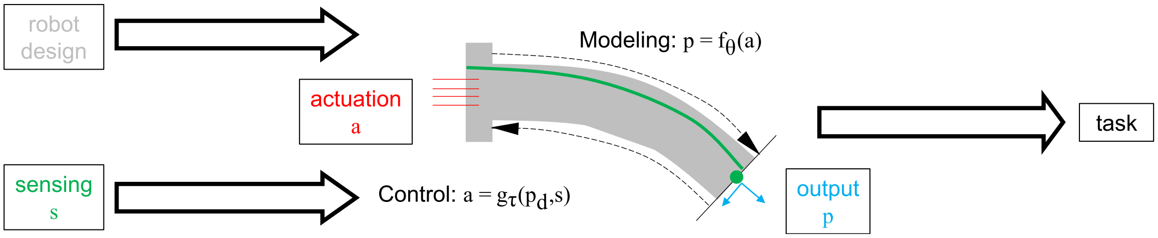


Fig. 1. Diagram of soft robot modeling and control processes. Robot design (grey) provides a specific soft robot structure and actuation pattern. The sensing system (green) obtains robot information via the sensors of such a robot. Modeling f_θ is a function that predicts the robot state p (blue), as end position and robot pose, according to the actuation variables a (red.) Control g_τ aims to decide actuation variables a with the desired state p_d and sensing inputs s. Finally, such a robot system can achieve a variety of tasks. θ and τ represent the parameters in the data-driven methods of modeling and control, respectively.

complex than that of their rigid counterparts, which also leads to challenging control tasks [13], [14]. Therefore, modeling and control play essential roles in soft robot research.

The working process of soft robots is summarized in Fig. 1. Various kinds of robot designs are used in soft robots, like 1 DOF soft fingers [15], 3 DOF continuum robots [16], parallel robots [17], concentric tube robots [18] and so on. To actuate a soft robot, the researchers apply actuation approaches like fluid-driven methods [19], cable-driven method [20], electroactive polymers [21], shape memory alloy (SMA) [22], etc. Also, there are multiple categories of sensors, e.g., optical markers [23], EM trackers [24], flex sensors [25] and Fiber Bragg Grating (FBG) [26], [27]. Some reviews detailedly introduce soft robots with mechanical aspects [28], [29]. Based on these hardware implementations, soft robot modeling can be seen as a function that takes actuation a as the input and robot state p, such as end position, orientation, and shape, as the output. Meanwhile, soft robot control is the inverse process of modeling, which takes the desired robot state p_d and the sensing signal s as the input and actuation a as the output. The input and output choices may change considering the control strategy. For example, the open control strategy in [30] only requires the target positions p_d for the trajectory following, but the close loop control approach in [23] utilizes both the target position p_d and the previous end positions from the sensing system s. Each mapping requires a model, f or g, and the parameter θ or τ, which are partly influenced by robot design and sensing system. Finally, such control methods are leveraged for simple tasks, like target reach [31] and trajectory following [32], and challenging tasks, like interaction adaptation [3] and navigation [33].

Due to the variety of structures and the complexity of behaviors, it is challenging to propose accurate physical models and corresponding control strategies for every soft robot. However, data-driven methods have shown considerable benefits. Data-driven approaches summarize the features of data collected from robot motions without the necessity of robot design knowledge. Furthermore, compared with physical approaches based on simplifications and hypotheses, data from real robots can illustrate the real features of soft robots. Thanks to these advantages, some data-driven approaches can be proposed for robot modeling and control with optimization [34] or learning [19]. Data-driven approaches can be applied for various kinds of modeling and control strategies, like kinematics modeling [35], dynamic modeling [36], open loop control [37], and close loop control [38]. Moreover, they can also be utilized for observer [39] which includes modeling as prediction and sensing characterization [40] which can be applied in control.

This review aims to present and compare data-driven methods applied to soft robot modeling and control in parallel. Although some reviews are related to soft robot modeling and control [14], [41], [42], [43], they do not introduce modeling and control simultaneously. In this review, first we briefly introduce foundations for data-driven approaches, which are physical models and the Jacobian matrix. Physical models provide simulation environments and insight into soft robots, while the Jacobian matrix infers the data relationship for data-driven approaches. Then, we classify data-driven methods applied to soft robot modeling and control into three categories: statistical method, neural network, and reinforcement learning. We introduce the corresponding modeling and control approaches for each kind of data-driven approach parallelly and compare them within and among these categories. Finally, conclusions on data-driven approaches applied to soft robot modeling and control are proposed, and we discuss the challenges and emerging directions of this area. Fig. 2 shows the paper number of different methods cited in this review according to the publishment time. In Fig. 2-(a), one paper may fall into multiple categories if it involves several methods, but one paper only falls into one category considering the main method in this paper in Fig. 2-(b). Fig. 3-(a) and Fig. 3-(b) summarize the number of papers according to the publishment time and method category, respectively. The summary of data-driven method categories is shown in Table I.

II. Foundations for Data-Driven Approaches

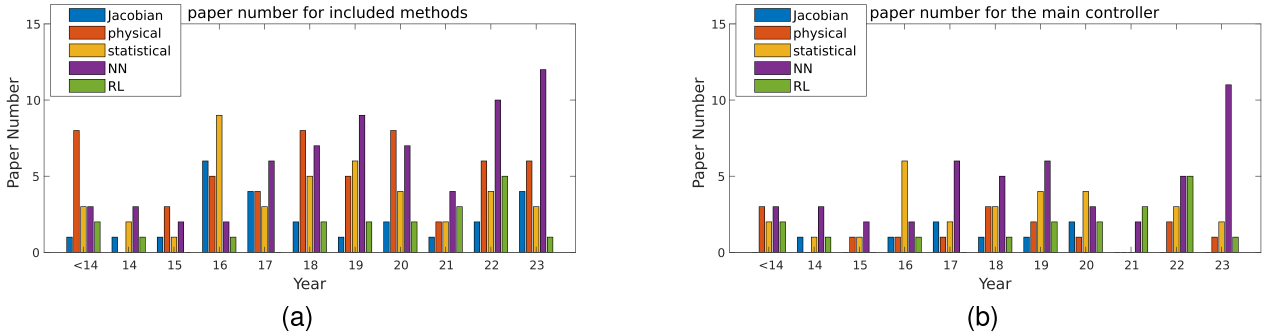
In this section, we introduce two significant foundations for data-driven methods, which are physical models and the Jacobian matrix. Both of them are very popular and effective in rigid robot modeling and control. In this case, the researchers also try to employ them in soft robots. They stimulate the development of data-driven approaches by providing simulators or inferring data relationships.

A. Physical Models

Physical models are significant in soft robotics because they can illustrate the nature of soft robot motion and deformation. Also, they are fundamental to data-driven methods by providing simulation environments and data. Discretization methods

TABLE I
SUMMARY OF METHOD CATEGORIES

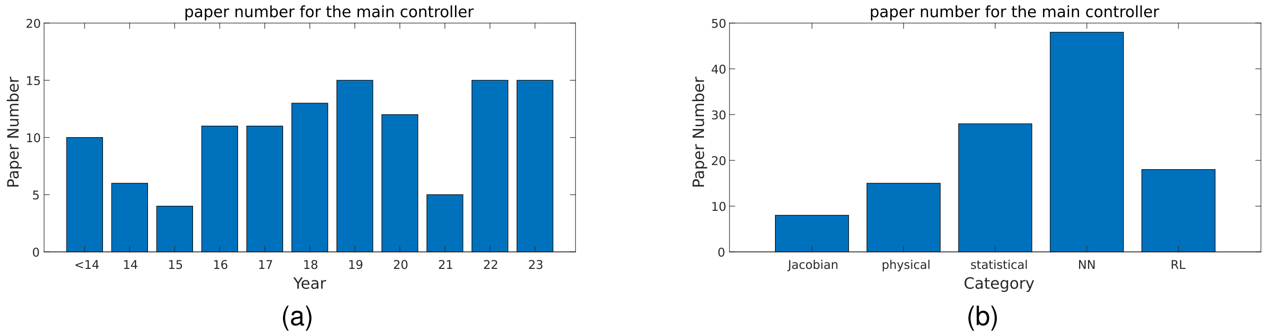
Method	Data amount	Modeling	Controller	Online updating	Dynamic control	Control feature
Statistical model	A moderate amount	Forward models (GPR, GMM, etc.)	Most models (GPR, GMM, etc.) and optimization based on forward models	✓	✓	Extract features only from data, forward and inverse mapping with the same model
NN	A large amount	Forward models	Models similar to modeling networks	✗	✓	Various structures and tasks
RL	A huge amount	Explored environment	Agent	✗	✓	Exploring environment and exploiting data, high level tasks



(a)

(b)

Fig. 2. Paper number for (a) included methods and (b) the main method.



(a)

(b)

Fig. 3. Paper number for the main method according to (a) years and (b) categories.

like FEM have been applied for robot simulation. The Cosserat model and its discretization method, like geometric variable-strain (GVS), provide simulators specifically for soft robots. Piecewise constant curvature (PCC) can be seen as a special case of GVS and is widely applied to the soft robot simulation based on the assumption of deformation shape. Some specific models, like the pseudo-rigid model and the concentric tube model, are proposed for specific soft robots. Some typical physical models are shown in Fig. 4. In robot control, methods considering physical models or parameters are model-based, while approaches only employing data and not using physical relationships inherent in robots can be seen as model-free [44].

FEM is a popular discretization simulation method in robotics. This method can be applied to various soft robots, for example, soft foam robot hand [45], soft pneumatic actuator [46], and PneuNet bending actuator [47]. This model can provide high-accuracy simulation with the requirement of material parameters such as mass density, Young's modulus, and Poisson's ratio. Generally, FEM works as an offline modeling method, such as producing a dataset [46] for NN learning and building an exploring environment for RL [45]. FEM is applied in the simulator SOFA [48] for deformation and interaction simulation. Moreover, a series of works [17], [49], [50] focus on FEM applications for real-time control. In order to cope with this highly complicated model, methods like local linearization [17] and reduced-order control model [50] are employed for the trade-off between the model accuracy and control frequency.

Besides the discretization of 3D small elements, the discretization of 1D Cosserat rod is also applied. Cosserat rod models soft continuum robots with a series of rigid cross-sections, and includes bend, twist, stretch, and shear. The simulator PyElastica [51] applies a discrete geometric form of the Cosserat rod model for soft robot simulation. The most general strain-based discretization method of the Cosserat rod is GVS [52], which discretizes the continuous Cosserat rod model into a finite set of strain basis functions. SoRoSim [53] integrates GVS for soft, rigid, and hybrid robotic system simulation. PCC is a special GVS, which only includes bending and simulates a continuum robot with several curves in some bending angles [54]. Original PCC only shows the geometrical information and dynamic PCC models include spatial dynamic

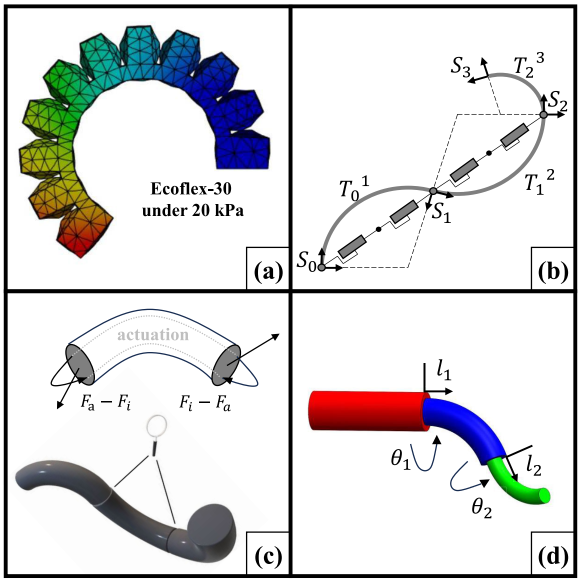


Fig. 4. Diagrams of (a) FEM, (b) PCC with pseudo-rigid model, (c) Cosserat rod model, and (d) concentric tube model. Element motion and deformation in (a) represent the soft robot deformation. The grey soft robot in (b) is represented by a series of constant curves, and the augmented rigid robots model the soft robot dynamically. The Cosserat rod model in (c) includes forces and moments inherently. The concentric tube model in (d) represents the robot motion with the configurations l_1, l_2, θ_1 and θ_2.

TABLE II
Physical Model Paper Comparison

Publication	Simulator	Simulation approach	Supplementary introduction
Faure *et al.* [48]	SOFA	FEM	Widely applied in organ simulation
Gazzola *et al.* [51]	PyElastica	Cosserat rod	Validated with soft filaments
Mathew *et al.* [53]	SoRoSim	GVS	Integrated into a user-friendly MATLAB toolbox
Iyengar *et al.* [31]	Customized	Concentric tube model	RL is applied for control

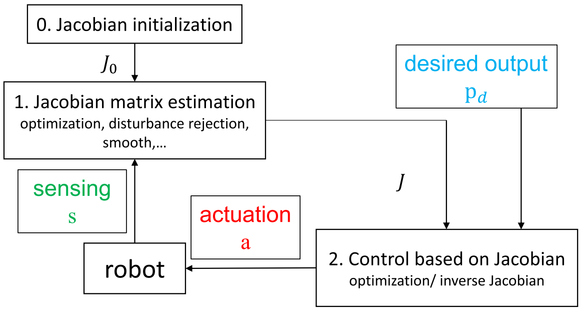

0. Jacobian initialization
J_0
1. Jacobian matrix estimation
optimization, disturbance rejection, smooth,...
desired output
p_d
J
sensing
s
actuation
a
robot
2. Control based on Jacobian
optimization/ inverse Jacobian

Fig. 5. Diagram of the Jacobian matrix estimation and control. Based on the Jacobian matrix initialization, the Jacobian matrix is updated based on the sensing s. Then, the estimated Jacobian matrix is applied for control.

effects [55] and external interactions [56]. Of note, PCC is a simplification of the nonlinear Euler-Bernoulli beam, which can be applied for soft robot modeling under external interaction without the constant curvature assumption [57].

There are also some physical models for specific kinds of soft robots. Considering concentric tube robots, the authors of [18] leverage rotations and translations of tubes as actuation variables for the concentric tube robot model, as shown in Fig. 4-(d). The control tasks are achieved by finding the inverse kinematic solutions. Neural networks are utilized for both modeling and control in [58] based on this model. The concentric tube model has also been used for RL control in [31] to provide action variables. Pseudo-rigid models can approximate soft robots with rigid counterparts [59] as shown in Fig. 4-(b), and the sophisticated motions of soft robots are simulated with the help of pseudo springs [60] and dampers [61].

In conclusion, the research of physical models deepens the understanding of soft robot nature and produce various soft robot simulators for data-driven approaches. A comparison of some typical papers applying physical models is shown in Table II. This table includes the simulator SOFA based on FEM [48], the simulator PyElastica based on Cosserat rod [51], the simulator SoRoSim based on GVS [53], and RL works using the specific model concentric tube model [31].

B. Jacobian Matrix

The Jacobian matrix can infer the relationship between the actuation and position velocities by model linearization. Thanks to its concision and effectiveness, the Jacobian matrix is widely applied in rigid robots, whose explicit physical models are easy to propose. Also, it is possible to linearize such models. However, due to the difficulties of building accurate and explicit models for soft robots, optimization is utilized for Jacobian matrix estimation instead of directly linearizing explicit models. The included robot model is just a general model like $p = f_\theta(a)$ instead of an explicit and physical one, and it is only employed for illustration instead of calculation [34].

The Jacobian matrix estimation and control process is summarized in Fig. 5. The first work applying the Jacobian matrix in the main approach is [34]. For Jacobian matrix initialization, each actuator should be actuated with a small incremental amount in order to estimate the initial Jacobian matrix J_0 [34], [62], [63], [64]. Then, the Jacobian matrix is updated according to the actuation and end position change in the last step. The matrix update strategy can be shown as

$$\begin{aligned} \min_{\hat{J}^{k+1}} \quad & \|\triangle \hat{J}\|_2 \\ s.t. \quad & \triangle x^k = \hat{J}^{k+1} W \triangle y^k \\ & \hat{J}^{k+1} = \hat{J}^k + \triangle \hat{J} \end{aligned} \tag{1}$$

where $\triangle x^k$ and $\triangle y^k$ are the end effector displacement and actuation change at step k, respectively. $J = [J_1 \ J_2 \ \dots \ J_n]$ is the Jacobian matrix and n is the dimension of the actuation variable y. $W = diag(\|J_1\|_2, \|J_2\|_2, \dots, \|J_n\|_2)$ is a weighting matrix and $\hat{J} = JW^{-1}$ is the unit Jaocbian matrix. $\hat{J}^k$ and $\hat{J}^{k+1}$ are the unit Jacobian matrix at step k and $k+1$.

The actuation variables are derived as the cost function in an optimization problem considering the constraints of the Jacobian matrix and target end position for control. The control strategy for step $k+1$ can be shown as

$$\begin{aligned} \min_{\triangle y^{k+1}} \quad & \|y^k + \triangle y^{k+1}\|_2 \\ s.t. \ & \triangle x_d = \hat{J}^{k+1} W \triangle y^{k+1} \end{aligned}$$

TABLE III
JACOBIAN MATRIX PAPER COMPARISON

Publication	Forward Model	Control method	Control target	Main controller	Control frequency	Jacobian matrix updating input	Controller details
Yip and Camarillo [34]	Jacobian matrix	Optimization	Position control	Jacobian model	1kHz	End effector positions and tension	Fast online controller
Yip and Camarillo [20]	Jacobian matrix	Optimization	Force and position control	Jacobian model	≥2kHz	End effector forces, positions and tension	Controller for position and force
Jiang *et al.* [69]	Jacobian matrix	Inverse Jacobian matrix	High-level tasks	RL	2Hz	End effector pose and pressure	Jacobian model applied as the low-level controller

$$y^k + \triangle y^{k+1} \geq y_{min}$$
$$y^k + \triangle y^{k+1} \leq y_{max} \tag{2}$$

where $\triangle x_d$ is the desired displacement for the end effector. The actuation value $y^k + \triangle y^{k+1}$ is constrained between the minimal and maximal actuation value y_{min} and y_{max} according to the robot structure. Except for the optimization control method in Eq. 2, the inverse Jacobian matrix is also utilized for control [39].

Considering nonlinearity and hysteresis of soft robots, the Jacobian matrix in the last step may not be accurate and suitable for this step. It is challenging for the original Jacobian estimation method to face complex tasks due to its oversimplification. Therefore, some researchers try to adapt this method for various applications. For example, the authors of [64] also emphasize that there may be a significant deviation between the estimated and real Jacobian matrix, and they alleviate this issue by smoothing the activation values. Force-displacement model is included for force control in [20]. When end effector and actuation displacement between only one step is involved in [34], the displacement among multiple steps is utilized in [65]. Also, the Jacobian matrix is adjusted in [66] by multiplying a rotating matrix considering the difference between the intended and measured motions. The fusion of sensing information from a camera and FBG provides accurate positions in [63] for stable and precise Jacobian matrix estimation.

Although so many adaptations have been proposed, the Jacobian matrix is still an initial solution for many labs and motivates applying the other approaches, especially data-driven ones. A neural network is applied for control in [67], which is based on the Jacobian matrix estimation process. A honeycomb pneumatic networks arm is controlled by the inverse Jacobian matrix in [68], and the same soft robot is included in [69] for challenging manipulation tasks. The latter work employs the Jacobian matrix in a low-level behavior controller and compares it with RL.

In summary, the Jacobian matrix is a concise and simple method for soft robots. The matrix is updated online with a high frequency in most cases, which can be achieved due to the simple structure. Meanwhile, the oversimplified linearization necessitates online updating and high control frequency. A comparison of some typical papers using the Jacobian matrix is shown in Table III. This table contains the first paper employing the Jacobian matrix for soft robot control [34]. Force control is achieved in [20] considering a force model in the control optimization problem. The authors of the last work [69] take inspiration from the Jacobian matrix in [68] and implement RL based on the same robot.

III. STATISTICAL METHOD

Statistical methods are utilized to build the mapping functions between different variable spaces with only data, and physical relationships among these spaces are unnecessary. For example, the authors of [70] infer the relationship between the actuation input and image feedback for kinematic control, and the authors of [21] plan the actuation variables based on the temporal values. There are many regression approaches utilized in soft robots, like linear regression [71], local weight regression (LWR) [72], support vector regression (SVR) [73], Gaussian process regression (GPR) [25], and local weight projection regression (LWPR) [16]. Other than the regression methods, the Gaussian mixture model (GMM) [74] summarizes collected data with a joint data distribution, and the extended Kalman filter (EKF) [39] estimates robot states as an observer.

A. Regression Method

Regression methods with different models are employed in soft robot modeling and control. These methods aim to fit the training data with a specific model, like a linear function or a Gaussian process, by optimizing the parameters and decreasing the loss. Then, the trained regression model can take some observation samples as model input and predict the corresponding values. For instance, linear regression, a simple regression approach, is applied in [71] to map FBG signals into soft robot end position for sensing. A linear function is included for fitting, and the parameter matrix is optimized. Similarly, LWR employs the linear function but considers the distance between the collected data and the observation samples for fitting. In [72], data from human demonstrations is used for fitting. The temporal value is taken as input to decide action for control.

SVR is utilized in [73] for the forward kinematic modeling and in [75] for the close loop position controller. It has been proven in [75] that SVR gets better approximation accuracy than NN on a simple function, but this model requires more convergence time than NN on a large amount of data (15625 samples) in [73], which may be caused by the different optimization strategies or mature NN optimization software. The SVR modeling and control algorithm in [75] can be summarized as

$$\begin{aligned} \min_{W,B} \ & \frac{1}{2}\sum_{j=1}^{M}\|w_j\|^2 + C\sum_{i=1}^{n} L(\mu_i), \\ s.t.\ & \mu_i = \|y_i - (\phi(x_i)W + B)\| \\ & W = [w_1, w_2, \ldots, w_M] \end{aligned} \tag{3}$$

where d, M are the dimensions of the mapping input x_i and output y_i, and n is the size of the learning dataset. The function $\phi(x_i)W + B$ aims to estimate the mapping output with a nonlinear transformation $\phi(\cdot) : R^{1\times d} \to R^{1\times h}$ and the optimized matrices $W \in h \times M$ and $B \in 1 \times M$. $L(\cdot)$ is a loss function and the extended Vapnik ϵ-insensitive loss function based on L2-norm is applied in [75]. C is the tradeoff parameter that adjusts the estimation errors and regularization. For kinematics modeling, SVR takes the actuation as input x and estimates y as the robot state. For control, which can be seen as the inverse process of modeling, SVR receives the desired displacement as input and decides the actuation.

GPR employs the Gaussian process, a probability distribution over functions, for fitting. For modeling, GPR predicts robot states, such as position or shape parameters, based on the training dataset and current actuation variables. For example, GPR is applied in [25] to predict the actuator curvature. Given N training inputs $X = [x_1, x_2, \ldots, x_N]^T \in R^{N\times i}$ and N curvatures as the training outputs $Y = [y_1, y_2, \ldots, y_N]^T \in R^{N\times 1}$, where i represents the dimension of each input x. The prediction process based on the test input x_t can be shown as

$$\begin{aligned} \mu(x_t) &= k_*(X, x_t)^T K(X, X)^{-1} Y, \\ \Sigma(x_t) &= k(x_t, x_t) - k_*(X, x_t)^T K(X, X)^{-1} k_*(X, x_t), \end{aligned} \tag{4}$$

where $\mu(x_t)$ and $\Sigma(x_t)$ are the predictive mean and variance. $k(\cdot,\cdot) \in R$ is the kernel function applied in GPR, a squared-exponential kernel function in most cases, and $k_*(X, x_t) = [k(x_1, x_t), k(x_2, x_t), \ldots, k(x_N, x_t)]^T \in R^{N\times 1}$. $K(X, X) \in R^{N\times N}$ is a covariance matrix with entries $K_{ij} = k(x_i, x_j), i, j = 1, 2, \ldots, N$. The above prediction process supposes that the prior mean is zero, and one can preprocess the sample output Y by zeroing the mean before fitting and prediction. It should be noticed that the noise of the mapping is considered in [70], which is assumed as white Gaussian noise with zero mean and variance σ_n^2. In this case, the prediction can be derived as

$$\begin{aligned} \mu(x_t) &= k_*(X, x_t)^T (K(X, X) + \sigma_n^2 I)^{-1} Y, \\ \Sigma(x_t) &= k(x_t, x_t) - k_*(X, x_t)^T (K(X, X) + \sigma_n^2 I)^{-1} k_*(X, x_t). \end{aligned} \tag{5}$$

With the forward model derived from GPR, the authors of [4] propose a control strategy by minimizing a cost function containing the predicted errors and actuation variables. GPR is also employed in close-loop kinematics control in [70] by predicting desired actuation variables based on the robot state feedback. The authors of [15] aim to predict the difference of robot states instead of only the next states as the modeling part in optimal control.

Based on LWR, which utilizes each training data as a local model, LWPR projects training data into several linear models and weighs them with Gaussian kernels. The covariances of these kernels are decided by an incremental gradient descent based on stochastic leave-one-out cross validation criterion [76], [77]. To find the optimization of the parameters in linear models, which is a redundancy problem, a null-space behavior is defined as a guide. This user-defined behavior is applied in a cost function and attracts the actual behaviors. For example, the robot elongation is minimized in [16] for a relatively straight shape, and the overall inflated chamber pressures are minimized during the control process in [78].

B. Gaussian Mixture Model

GMM encodes collected data into a data distribution composed of multiple Gaussian components. In the fitting step, GMM parameters are optimized to fit the collected data. During the prediction step, the input data works as the prior, and the posterior of GMM under some input data will be used as prediction output. Of note, once a GMM is built, each kind of data plays the same role in principle. In such a joint probability density function, every dimension can be applied as a prior and derive expectations on the remaining dimensions. In this case, GMM in [74] produces both forward kinematics modeling and position control strategy with actuation variables and desired end positions as the prior, respectively. The catheter kinematic GMM is represented by the joint probability density function:

$$P[p[k+1], p[k], a], \tag{6}$$

where $p[k]$ denotes the robot state at step k. The forward modeling process can be shown as

$$E[p[k+1] | p[k], a], \tag{7}$$

which is the conditional mean of the model Eq. 6 given the robot state $p[k]$ and actuation a. The control strategy is

$$E[a | p_d[k+1], p[k]], \tag{8}$$

which is the conditional mean of the model Eq. 6 given the robot state $p[k]$ and desired state $p_d[k+1]$.

In addition to modeling and control, such data encoding characteristic develops some planning solutions. For example, the authors of [21] encode pose and temporal value into a GMM for navigating through narrow holes based on human demonstration. Moreover, other task parameters like the rotation matrix of the coordinate system are included as the planning objects in the GMM of [79]. Making use of its encoding ability, GMM transfers demonstrations on rigid robots to the STIFF-FLOP continuum robot in [80].

C. Extended Kalman Filter

Considering one existing model and its prediction, the Kalman filter can be applied as an observer and corrects the predicted values with measurement. Due to the nonlinear responses from soft robots, most modeling process is nonlinear, and the extended Kalman filter (EKF), instead of the original linear Kalman filter, is widely applied in soft robots. P, Q, R represent the estimation covariance, process noise covariance, and measurement noise covariance, respectively.

In prediction, the state at the $k+1$ step is predicted as $p_{k+1|k}$ based on the state $p_{k|k}$ and actuation a_k at the k step.

$$\begin{aligned} p_{k+1|k} &= f(p_{k|k}, a_k), \\ P_{k+1|k} &= A_k P_{k|k} A_k^T + Q_k, \end{aligned} \tag{9}$$

where $f(\cdot,\cdot)$ represents the nonlinear forward modeling, A_k is the local linearization of $f(\cdot,\cdot)$. The correction process corrects the prediction state to $p_{k+1|k+1}$ considering measurement output s_k from sensing.

$$\begin{aligned} K_k &= P_{k+1|k} C_k^T (C_k P_{k+1|k} C_k^T + R_k)^{-1}, \\ p_{k+1|k+1} &= p_{k+1|k} + K_k(s_k - g(p_{k+1|k})), \\ P_{k+1|k+1} &= (I - K_k C_k) P_{k+1|k}, \end{aligned} \tag{10}$$

where $g(\cdot)$ represents the measurement process, C_k is the local linearization of $g(\cdot)$. K_k is the Kalman Gain and evaluates the reliability of measurement and prediction. EKF is commonly leveraged since it adjusts the modeling process and provides an accurate robot state estimation. For example, the authors of [81] map actuator variables and segment parameters into the robot pose to build the nonlinear forward modeling with the transformation matrices, and the sensing signal from position sensors is applied for correction. The authors of [82] aim to estimate robot poses and physical parameters and apply pose measurement for correction. NNs like wavelet networks [83] are also employed as the forward model in EKF for curvature angle estimation. In their following research, the authors of [84] also estimate the external force as the unknown system input based on the state estimation from a similar EKF. Similarly, the unknown external forces are taken as the system state p and accurately estimated in [85]. Furthermore, due to the modular structure of the snake robot in [86], the EKF is adapted by changing the dimension of state variables according to the advancing or retracting motion for shape estimation.

D. Summary Statistical Method

Because statistical models only consider data relationships, such models have shown high potential to cope with various kinds of data. At first, statistical models only include robot actuation for modeling [87]. Then, position feedback is involved in [75] for adaptive modeling and control. Furthermore, temporal values are applied for control in [21] and [72]. The authors of [70] even include visual information for kinematic control. Recently, sensing information from various sensors like resistance and force sensors has been employed in [84] and [88] for modeling adjustment. Most statistical approaches can be applied for both modeling and control with different inputs, like two SVR models in [75].

Besides data categories, the methods also evolve over time. For instance, the Jacobian matrix is involved in the EKF in [85], and the pseudo rigid robot model takes its place in [89]. The authors of [39] apply the adaptative Kalman filter, which shows strong robustness against the model nonlinearity and uncertainty instead of EKF. Recently, the unscented Kalman filter is leveraged in [88], which can apply the implicit Gaussian process for robot modeling. Also, GPR also evolves over time. The whole working space is divided into several regions, and each part requires a single GPR model for local modeling in [70]. Then, only one online learning GPR model is employed to model the whole working space in [15], and a meta-learning GPR model is employed in [90] for multiple new unknown working spaces.

Statistical methods make data distribution assumptions from the perspective of statistics. They can attain an acceptable performance even with a small amount of data and become more effective with more data. Moreover, most of them can be leveraged for both modeling and control. A comparison of some typical papers applying statistical methods is shown in Table IV. This table first includes a simple regression model SVR in [75], then includes two regression approaches GPR in [15] and LWPR in [16]. Finally, the observer EKF in [81] is introduced.

IV. Neural Network

Considerable efforts have been focused on NN applications in the soft robot field. In the early years, extreme learning machines (ELM) [91] and radial basis function (RBF) [35] were popular choices. Nowadays, researchers prefer multilayer perceptron (MLP) [92] and recurrent neural network (RNN) [93] due to their generalized and sequence-related structures, respectively. Moreover, for some special proposes like image processing, autoencoder (AE) [94] and convolutional neural network (CNN) [95] are utilized. Some typical NNs are shown in Fig. 6.

A. ELM and RBF

ELM only comprises an input layer and an output layer. The input layer weights and bias are randomly assigned before training and fixed during training, while the output layer weights are trained to decrease loss. A simple loss is utilized in [96] for kinematics control, which only aims to decrease the estimation error. The ELM is

$$\hat{a} = W^{out} \cdot f(W^{inp} \cdot p + B), \tag{11}$$

where $\hat{a}$, p denote the estimated actuation value and the robot state, W^{out}, W^{inp} denote the output and input layer weights, B denotes bias, and $f(x) = (1+e^{-x})^{-1}$ is the sigmoid activation function. The training process of the original ELM in [96] can be shown as

$$\begin{aligned} &\min_{W^{out}} \quad \|A - \hat{A}\|, \\ &s.t. \quad \hat{A} = W^{out} \cdot P, \end{aligned} \tag{12}$$

where A and P are all the real actions and input layer output in the training dataset, and $\hat{A}$ is the ELM estimation based on P. The optimized output layer weights are $\hat{W}^{out} = A \cdot P^{\dagger}$, where $P^{\dagger}$ is the pseudo-inverse of the input layer output P.

In addition to estimation error, the authors of [97] and [98] involve the norm of the output weight matrix into optimization error as a regularization term to avoid too large output weights for kinematics control. Constraints on the range of outputs are applied in the ELM controller [3] according to the constraints of the real actuation. The training process can be shown as

$$\begin{aligned} &\min_{W^{out}} \quad \sum_{i=0}^{N} \|a_i - \hat{a}_i\|^2 + \alpha \cdot \|W^{out}\|^2, \\ &s.t. \quad \hat{a} = W^{out} \cdot f(W^{inp} \cdot \mathbf{p} + B), \end{aligned}$$

TABLE IV

Statistical Method Paper Comparison

Publication	Modeling	Control	Control frequency	Data amount	Dynamic control	Online update	Controller details
Melingui *et al.* [75]	SVR	The other SVR model	-	4096	✗	✓	Adaptative controller by online controller updating
Tang *et al.* [15]	GPR	Optimization	10Hz	120	✓	✓	Initialization in simulation and learning in reality
Lee *et al.* [16]	LWPR	The same LWPR model	20Hz	-	✓	✓	Online controller to adapt distrubance
Ataka *et al.* [81]	EKF	Inverse Jacobian matrix	40Hz	-	✓	✓	Controller for segmented robot and obstacle avoidance

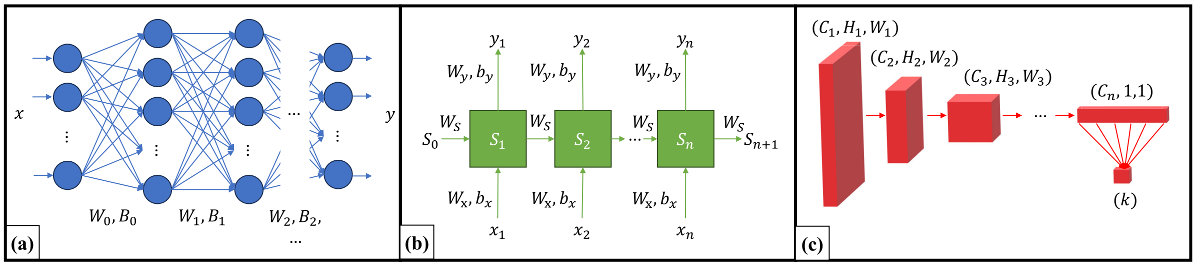


Fig. 6. Diagrams of (a) MLP, (b) RNN, and (c) CNN. MLP is composed of multiple layers. Parameters W_*, B_* in one layer are in parallel and can be trained simultaneously. RNN takes the input data x sequentially, and each bar shares the same weights W_S, W_x, W_y, b_x, and b_y. CNN obtains matrix with channel C_1, height H_1, and width W_1 as input. In the network feedforward, the channel number improves while the width and height decrease using kernels. Finally, CNN outputs a matrix with dimension $(C_n, 1, 1)$, and a fully connected layer is employed to map to the target dimension k.

$$\frac{\partial}{\partial p_i}\hat{a}(\mathbf{p}) > 0 : \forall \mathbf{p} \in \Omega, \quad i = 1, \dots, n$$
$$a_{min} < \hat{a}(\mathbf{p}) < a_{max} : \forall \mathbf{p} \in \Omega, \quad i = 1, \dots, n \qquad (13)$$

where a is the real actuation value, α is the regulation parameter, N is the size of the training dataset, n is the degrees of actuation, and Ω represents a set of samples basically cover the input space. The second condition guarantees that ELM has the same actuation direction as the real conditions, and the third is the actuation range constraint. Recently, ELM has been utilized for online pose estimation in [40] thanks to its simple structure and fewer parameters compared with other complicated NNs.

Based on the structure of ELM, RBF changes the activation function from a sigmoid activation function to multiple Gaussian functions in [35] for forward kinematic modeling. Clustering can be applied to the training dataset to select the center of the Gaussian functions [98]. Although ELM and RBF include some basic elements of NN, e.g., activation functions and neurons, they contain some designed constraints for specific tasks and waste a part of their potential mainly due to the fixed parameters.

B. MLP

MLP is the most popular NN. The diagram of MLP is shown in Fig. 6-(a). Generally, MLP can be shown as

$$y = W_n \cdot f(W_{n-1} \cdot f(\dots W_0 \cdot f(x) + B_0 \dots) + B_{n-1}) + B_n, \qquad (14)$$

where W_n, B_n represent network weight and bias of n-th layer, and y, x are the output and input of MLP. To model a robot, the input is actuation, and the output is the robot state. For control, the desired and real robot states are input, and MLP can provide the estimated actuation.

The first paper utilizing MLP in the soft robot field is [99], which is also the first paper utilizing NN. This work designs a particular parameter updating strategy for control instead of the backpropagation widely applied now. Similarly, training data is normalized in [92] with principal component analysis (PCA) instead of batch and layer normalization, which are widely employed currently. MLP has many hyperparameters and changeable components, and plenty of papers have been conducted on them. For example, the authors of [100] compare the performance of MLPs composed of different neuron numbers on position control tasks. Three activation functions (i.e., log-sigmoid, linear, and tan-sigmoid) are tested in [47] for pneumatic robot modeling. The stochastic gradient descent (SGD) optimizer is applied in [101] instead of the popular Adam optimizer. The authors of [24] comprehensively investigate the influence of hidden layer number, neuron number, and batch size on training time and validation loss of sensing tasks. In general, there is no optimal combination of the hyperparameters and component structures for all tasks, but most choices can obtain acceptable performance on most tasks. The detailed influence of each hyperparameter on the MLP performance has yet to be fully explored and explained.

Some uncommon MLPs have been applied to soft robots. The authors of [22] provide targets to the hidden neurons of the MLP directly, similar to the conditional generative adversarial network (CGAN), which takes advantage of the information that is indirectly related to the mapping but restricts the potential of NN. A U-Net-like MLP is leveraged in [102] for robot modeling in model predictive control (MPC), which connects the former and latter layers sequentially. Multiple MLPs are connected in [103], which have different applications, e.g., forward kinematics modeling in simulation and sim-to-real transfer learning. The combined MLP can collect actions in simulation and estimate the corresponding robot states.

For soft robot modeling and control, some unique features of MLP are developed. For example, the inputs of some MLPs take physical models into consideration. Based on the concentric tube model, the translations and rotations of the concentric tubes are employed into MLP inputs in [104] for modeling. Furthermore, the authors of [105] compare different joint space forms of the concentric tubes as input on the forward modeling estimation tasks. Similarly, PCC is included in [54] by taking curvatures and curve lengths as MLP input. The authors of [106] utilize MLP to estimate physical parameters like mass inertia matrix, which is a model-based approach with NN. Due to the time-delayed motion, end positions in the current and past timesteps are fed into the MLP in [107] for control.

C. RNN

RNN is a kind of NN that is designed specifically for sequential data. The diagram of RNN is shown in Fig. 6-(b). Although MLP can also receive temporal data, it is fed into networks simultaneously and fails to infer the sequential relationship. RNN takes data in sequence using recurrent structure. The n-th bar in Fig. 6-(b) can be represented as

$$\begin{aligned} S_n &= f(W_x \cdot x_n + W_s \cdot S_{n-1} + b_x), \\ y_n &= g(W_y \cdot S_n + b_y), \end{aligned} \tag{15}$$

where W_*, b_* are the RNN weight and bias parameters, S_n is the hidden state of the step n, and $f(), g()$ are activation functions. As shown in Fig. 6-(b), the network of each step takes the hidden state from the previous step. Such a structure infers the sequential relationship of data while the other NNs take the data input simultaneously. In this case, RNN is more suitable than the other networks like MLP and CNN for soft robots, which provide delayed responses. The modeling and control with RNN share the same data category requirement with MLP mentioned in Eq. 14, but RNN needs data from multiple time steps instead of a single time step.

The first RNN applied is modified Elman NN in [108], which restores information in previous steps with context nodes for dynamic control. Then, researchers leverage a nonlinear autoregressive network with exogenous inputs (NARX) in a series of papers [19], [67], [93] for dynamic control. This kind of RNN receives outputs in previous steps as a part of input in the current step.

Long short-term memory (LSTM) is a kind of RNN proposed for long-term dependence issues. This network has been used on endoscopic robot distal force prediction [109], [110] and external force position and magnitude prediction [111]. Moreover, such a network can cope with sensing signals from nonlinear time-variant soft sensors and achieves tasks like position prediction [112], object recognition [113], and shape reconstruction [94]. In addition to perception, the authors of [32] dynamically control a robotic catheter with LSTM to decrease contact force. The authors of [114] employ LSTM as an offline dynamic controller to cope with the nonlinear behaviors. In summary, RNNs perform satisfactorily on various tasks due to their sequential structures and memory ability.

D. Special NN

There are some NNs that take advantage of visual information in soft robot control. For example, AE is utilized to extract features from the images of soft robots in [94] and estimate the robot's shape. CNN estimates shapes [115] and joint values [116] based on robot images. The diagram of CNN is shown in Fig. 6-(c). Also, CNN can predict the orientation of the placenta in [95] and encodes robot deformation for shape estimation with the help of markers inside the chamber in [117]. Furthermore, although RNN is the most popular choice for sequential information processing, CNN can also be applied for dynamic modeling using rearranged pressure inputs [118]. The space relationship in a matrix is leveraged to infer the sequential relationship of an actuation sequence for hysteresis modeling. A 3D NN is employed in [119] for a segmental surgical manipulator, which considers the time sequence between layers and the segmental sequence between neurons for planning. Spatial sequences of soft modular robots are considered in [120] by utilizing bidirectional RNN. A generative adversarial network (GAN) is utilized for synthetic data in [121].

E. Summary Neural Network

The application of NN in soft robots started with [99]. First, some simple networks like ELM and RBF are included. Then, with the development of NN, the researchers give up the constraints of ELM, change the activation function of RBF, and enlarge the network. In this case, MLP becomes a good tool for both modeling and control. Furthermore, due to the hysteresis of soft robots, RNN is applied to deal with time-related data. AE and CNN are employed to process images.

To summarize, owing to the large variety of structures, NN is attractive to soft robot research. For most issues in soft robot modeling and control, it is highly possible to find a related NN solution. However, such models require a large amount of data due to their complicated structures, and it is challenging to update them online. A comparison of some typical papers applying NNs is shown in Table V. This table begins with a simple NN, RBF [122]. Then, a common NN, MLP [24], is included in this table. Finally, LSTM for control [32] and CNN for modeling [118] are summarized in Table V.

V. Reinforcement Learning

RL copes with high-level tasks by exploring the environment and exploiting data collected during exploration. This strategy trains an agent for complicated tasks and requires a long learning time and a massive amount of data. The agent is trained considering defined reward functions. This approach cannot be used for modeling. Moreover, it requires exploring environments, which may be provided by modeling methods.

A. RL Implementation

In the early years, statistical models were applied as the agents in RL instead of NN. A GPR model named Gaussian

TABLE V

Neural Network Paper Comparison

Publication	Modeling	Control	Hidden layer	Activation function	Neuron	Data amount	Dynamic control	Controller details
Melingui *et al.* [122]	RBF	The other RBF	1	Gaussian function	72/74	4096	✗	Forward and inverse model with the similar NN structure
Ha *et al.* [24]	MLP	N/A	1-4	ReLU	15-720	28738	✗	Hyperparameter finetuning
Wu *et al.* [32]	N/A	LSTM	4	Tanh/Sigmoid	128	260269	✓	Dynamic controller and minimizing contact force
Zhang *et al.* [118]	CNN	N/A	4	Tanh	$2 \times 2 \times 4$	3800	✓	CNN dynamic modeling with transformed data

TABLE VI

Reinforcement Learning Paper Comparison

Publication	Modeling	Control	Agent	Episode	Learning strategy	Training environment	Testing environment	Controller details
Engel *et al.* [123]	FEM	RL	Gaussian process	3000	GPTD	Sim	Sim	The first data model applied for soft robot control
Ji *et al.* [127]	CC	RL	NN	100	MADQN	Sim and Real	Sim and Real	Agent composed of evaluation and target network
Satheeshbabu *et al.* [37]	N/A	RL	NN	5000	DQN	Real	Real	Discrete task space
Li *et al.* [133]	CC	RL	NN	500	DDPG	Sim	Sim and Real	Pre-training in Sim and learning in Real

process temporal difference method is employed in [123] to control an octopus arm. As the RL agent, a GMM is trained to estimate robot shape and contact in [124] and control a flexible surgical robot to go through a tube in [125]. For robotic catheter control inside a narrow tube, a joint probability distribution is learned considering various variables like tip and entrance points, touch state, and action in [33]. Q-learning is implemented with a Q table or Q function as the agent. The agent decides the action A based on the state S and a certain policy like ϵ greedy. The training process of the Q table or function can be shown as

$$Q(S_t, A_t) \leftarrow \alpha[R(S_t, A_t) + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] + Q(S_t, A_t), \quad (16)$$

where S_t is the state at time t, A_t is the action decided by the Q table or function according to the state S_t, and $R(S_t, A_t)$ is the reward function. α and γ are the learning rate and discount rate, respectively. $\max_a Q(S_{t+1}, a)$ means the maximal Q value for the state S_{t+1} and every possible action a. For example, the authors of [69] exploit Q-learning for many sophisticated control tasks like turning a handwheel, unscrewing a bottle cap, drawing a line with a ruler, and so on.

With the help of NN as an agent, RL not only achieves simple tasks like position reach [126] and trajectory following [127] but also addresses some complex issues like gait design [128]. A soft snake robot is controlled to move on the ground and arrive at target positions in [128]. It is challenging to control the robot gait with traditional control methods, but RL is utilized for gait design and obstacle avoidance in [129]. The authors of [130] fuse the visual and shape information with NN in RL and control a flexible endoscopy to navigate.

B. RL in Soft Robots

Soft robots can take advantage of various unique RL strategies specifically for soft robots to cope with some challenges. The infinite DOFs lead to wide actuation and state spaces, which bring a burden to explore the whole environment. In this case, most researchers discretize these spaces. The action space in [123] is restricted to only six available actions, and the authors of [37] discretize the workspace into a 3D grid with a resolution of 0.01 m. Although discretization limits the RL potential, it produces simple spaces and decreases the training time. The soft robot in [131] is able to keep the end position invariant while changing the orientation with the help of RL.

One of the most considerable challenges of RL is exploring the real world, which has a high time cost and may damage robots. Therefore, modeling in simulation, especially with physical methods, is widely utilized in the training process. The authors of [132] first train an RL strategy to control a robotic catheter system in a simulator named CoppeliaSim and then test it on the real robot. Constant curvature (CC), a soft robot modeling method, provides a simulation environment for [133] at first, and the NN agent continues to learn in the real world using the Deep Deterministic Policy Gradient method (DDPG). In most cases, the explored environment is modeled by a trained NN. For example, MLP is applied for modeling in [134] as the exploring environment. Also, RNNs are utilized in [23], [38], which are NARX and LSTM respectively, for forward modeling of segmented pneumatic robots. Then, RL agents are trained and validated in reality.

C. Summary Reinforcement Learning

RL application in soft robots first starts with the help of statistical models as agents, and then NNs take the place due to their generalization. Discretization is widely applied for RL in soft robots. Meanwhile, RL leverages soft robot simulators and stimulates their development. With such RL strategies, now soft robots can achieve some complicated tasks.

Compared with other approaches, RL requires the most enormous amount of data. More critically, a predefined agent

TABLE VII
SUMMARY OF PAPERS AFTER 2019

Data model	Category	Publications
Statistical method	SVR	[73]
	kNN	[26]
	KF	[66], [84], [88]
	Gaussian Distribution	[111]
	GPR	[4], [15], [25], [40], [77], [88], [90], [133]
NN	ELM	[40]
	MLP	[24], [27], [30], [31], [38], [63], [73], [101], [103], [104], [106], [107], [116], [117], [126], [127], [132]–[134]
	RBF	[73]
	LSTM	[32], [110], [111], [113], [114]
	Special NN	[26] (AE), [102] (U-Net), [128], [129] (CPG), [115]–[117], [130] (CNN), [121] (GAN), [120] (biRNN)
RL	Q-learning	[69], [131]
	DQN	[132], [134]
	MADQN	[127]
	PPO	[128], [129]
	TRPO	[38]
	DDPG	[126], [133]

and interaction with the environment are necessary. Following such a high cost, RL can fulfill complex and high-level tasks. With some adaptation strategies like discretization and simulation transfer learning, the time and resource costs can be reduced to some extent. A comparison of typical papers applying RL is shown in Table VI. This table begins with an RL strategy applying the statistical model Gaussian process as agent [123], and the authors of [127] apply NNs for RL. Then, two strategies for soft robot RL are included. RL strategies like discretization are employed in [37], and the authors of [133] pretrain the agent in simulation and test it in reality.

VI. Conclusion and Discussions

In this section, we summarize the foundations, data-driven methods, and their representative papers in Subsection VI-A. The benefits and limitations of data-driven approaches involved in the review are included and compared in Subsection VI-B. Finally, we forecast the emerging directions for soft robot research in Subsection VI-C.

A. Summary

This review summarizes the data-driven approaches applied in soft robot modeling and control. The physical approaches provide simulators for data-driven methods, like SOFA [48], Pyelastica [36], and SoRoSim [53]. The Jacobian matrices describe soft robots and control the robots with the inverse Jacobian matrix or optimization. The authors of [34] firstly utilize the Jacobian matrix in the main approach.

Statistical models aim to achieve modeling and control with datasets. Regression methods like GPR [70] and LWPR [16] can estimate the mapping functions, and each trained model can be applied for either modeling or control according to the input of the training data. GMM [74] encodes the dataset into a joint data distribution, which can be applied for both modeling and control. Observers like EKF [88] can estimate robot states based on one existing modeling method. Recently, NN has been the most compelling tool for soft robots. ELM [3] was a popular choice in early years, and now MLP [100], which is more complicated, and RNN [19], which can cope with time-related data, have shown their benefits. Similar to statistical models, NN can also be applied for both modeling and control. RL shows good performance on position control [38], planning [31], and even some sophisticated manipulation tasks [69]. Generally, RL does not provide a robot model but exploits an existing environment. The papers based on data-driven methods, which are cited in this review and published after 2019, are summarized in Table VII.

It is apparent that sophisticated models like RL require a larger amount of data while achieving better performance, but they also improve the computation cost. Oversimplified approaches like the Jacobian matrix are only feasible for limited simple tasks, but they are easy to understand and can achieve a high control frequency. Considering both cost and performance, each model has its own advantages and disadvantages, and there is no optimal solution for all tasks.

B. Advantages and Limitations

Statistical modeling and control approaches are moderate and flexible approaches in these methods. Generally, they only require a moderate amount of data, which is less than the NN requirement but more than the requirement of the Jacobian matrix. Also, the control frequency is lower than that of the Jacobian matrix but higher than the RL frequency. One statistical model can be applied for both modeling and control online, but some models are only local models and lack generalization ability, which can be proven by the working space segmentation in [70]. NN is very suitable for soft robot modeling and control due to the nonlinear activation functions and complex network structure. This model shows its general applicability. However, the data amount requirement accelerates the aging of soft robots and takes a long time for data collection. NN cannot be a fast and online approach due to the slow training process. By exploring the environment and exploiting the data, RL can achieve some high-level tasks like navigation. Careful planning is not required even for sophisticated tasks. Meanwhile, RL has a high requirement for time and computation resources. Exploring the real world may damage the robots, and some environments, like the human body, cannot be applied for exploring. Highly realistic simulators are required if the RL agent is trained in simulation. To summarize, RL is a useful but consuming approach.

C. Emerging Directions

In the other research areas like computer vision and natural language processing, the original NN is simple at first [135]. Then, more complex NNs with large sizes and different structures are proposed, like ChatGPT, BERT [136] and YOLO [137]. Similarly, the size and complexity of NN in soft robots will improve. Such models can achieve more challenging tasks compared with simple NNs. However, large models lead to low frequency for control implementation, and one should consider balancing the model complexity and computation cost based on the modeling and control requirement.

The research on soft robots begins with the application of one single approach and simple tasks. Recently, there have been some papers combining multiple methods to solve difficult problems like model mismatch. For example, the authors of [114] apply offline RNN and online kinematics model for control. MPC and iterative learning controller are combined in [60]. Two NNs are included in [138] for RL agent and model mismatch adjustment. As discussed in Subsection VI-B, each method shows its own advantages and disadvantages. The usage of multiple methods can take advantage of every approach and achieve better performance.

The medical environment, as one of the most significant applications of soft robots, has a high standard for safety, efficiency, and convenience. To involve soft robots in medical applications, some advanced modeling and control strategies are required. Although so many works achieve controlling the robot end pose, it is challenging to control the whole robot shape and avoid contact which may damage the human body. NN and RL lack interpretability and are challenging to apply in real surgery. Also, it is impossible for RL to explore *in vivo*, and RL agents can only be trained in simulation or physical simulators. The automatic medical soft robot control is still in its nascence from the aspect of safety and data requirements. Cooperation among robotics researchers, doctors, and related departments is required to address these issues.

References

[1] M. Cianchetti, T. Ranzani, G. Gerboni, I. De Falco, C. Laschi, and A. Menciassi, "STIFF-FLOP surgical manipulator: Mechanical design and experimental characterization of the single module," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3576–3581.

[2] Y. Ansari, M. Manti, E. Falotico, M. Cianchetti, and C. Laschi, "Multiobjective optimization for stiffness and position control in a soft robot arm module," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 108–115, Jan. 2018.

[3] J. F. Queißer, K. Neumann, M. Rolf, R. F. Reinhart, and J. J. Steil, "An active compliant control mode for interaction with a pneumatic soft robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 573–579.

[4] Z. Q. Tang, H. L. Heung, X. Q. Shi, K. Y. Tong, and Z. Li, "Probabilistic model-based learning control of a soft pneumatic glove for hand rehabilitation," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 2, pp. 1016–1028, Feb. 2022.

[5] J. Tolvanen, J. Hannu, and H. Jantunen, "Stretchable and washable strain sensor based on cracking structure for human motion monitoring," *Sci. Rep.*, vol. 8, no. 1, p. 13241, Sep. 2018.

[6] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario, "Soft robot arm inspired by the octopus," *Adv. Robot.*, vol. 26, no. 7, pp. 709–727, Jan. 2012.

[7] S. Joe, M. Totaro, H. Wang, and L. Beccai, "Development of the ultralight hybrid pneumatic artificial muscle: Modelling and optimization," *PLoS ONE*, vol. 16, no. 4, Apr. 2021, Art. no. e0250325.

[8] A. Menciassi, S. Gorini, G. Pernorio, L. Weiting, F. Valvo, and P. Dario, "Design, fabrication and performances of a biomimetic robotic earthworm," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Aug. 2004, pp. 274–278.

[9] 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.

[10] J. Sun, L. Bauman, L. Yu, and B. Zhao, "Gecko-and-inchworm-inspired untethered soft robot for climbing on walls and ceilings," *Cell Rep. Phys. Sci.*, vol. 4, no. 2, Feb. 2023, Art. no. 101241.

[11] K. C. Galloway et al., "Soft robotic grippers for biological sampling on deep reefs," *Soft Robot.*, vol. 3, no. 1, pp. 23–33, Mar. 2016.

[12] C. Firth, K. Dunn, M. H. Haeusler, and Y. Sun, "Anthropomorphic soft robotic end-effector for use with collaborative robots in the construction industry," *Autom. Construct.*, vol. 138, Jun. 2022, Art. no. 104218.

[13] K. Chin, T. Hellebrekers, and C. Majidi, "Machine learning for soft robotic sensing and control," *Adv. Intell. Syst.*, vol. 2, no. 6, Jun. 2020, Art. no. 1900171.

[14] J. Wang and A. Chortos, "Control strategies for soft robot systems," *Adv. Intell. Syst.*, vol. 4, no. 5, May 2022, Art. no. 2100165.

[15] Z. Q. Tang, H. L. Heung, K. Y. Tong, and Z. Li, "A probabilistic model-based online learning optimal control algorithm for soft pneumatic actuators," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1437–1444, Apr. 2020.

[16] 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, Dec. 2017.

[17] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3982–3987.

[18] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 209–225, Apr. 2010.

[19] T. G. Thuruthel, E. Falotico, M. Manti, and C. Laschi, "Stable open loop control of soft robotic manipulators," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1292–1298, Apr. 2018.

[20] M. C. Yip and D. B. Camarillo, "Model-less hybrid position/force control: A minimalist approach for continuum manipulators in unknown, constrained environments," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 844–851, Jul. 2016.

[21] H. Wang, J. Chen, H. Y. K. Lau, and H. Ren, "Motion planning based on learning from demonstration for multiple-segment flexible soft robots actuated by electroactive polymers," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 391–398, Jan. 2016.

[22] C. Cheng, J. Cheng, and W. Huang, "Design and development of a novel SMA actuated multi-DOF soft robot," *IEEE Access*, vol. 7, pp. 75073–75080, 2019.

[23] 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.

[24] X. T. Ha et al., "Shape sensing of flexible robots based on deep learning," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 1580–1593, Apr. 2023.

[25] J. Jung, M. Park, D. Kim, and Y.-L. Park, "Optically sensorized elastomer air chamber for proprioceptive sensing of soft pneumatic actuators," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2333–2340, Apr. 2020.

[26] X. T. Ha et al., "Contact localization of continuum and flexible robot using data-driven approach," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6910–6917, Jul. 2022.

[27] Z. Dong et al., "Shape tracking and feedback control of cardiac catheter using MRI-guided robotic platform—Validation with pulmonary vein isolation simulator in MRI," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 2781–2798, Oct. 2022.

[28] C. Lee et al., "Soft robot review," *Int. J. Control, Autom. Syst.*, vol. 15, no. 1, pp. 3–15, Feb. 2017.

[29] K. Liu, W. Chen, W. Yang, Z. Jiao, and Y. Yu, "Review of the research progress in soft robots," *Appl. Sci.*, vol. 13, no. 1, p. 120, Dec. 2022.

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

[31] K. Iyengar, S. Spurgeon, and D. Stoyanov, "Deep reinforcement learning for concentric tube robot path following," *IEEE Trans. Med. Robot. Bionics*, vol. 6, no. 1, pp. 18–29, Feb. 2024.

[32] D. Wu et al., "Deep-learning-based compliant motion control of a pneumatically-driven robotic catheter," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 8853–8860, Oct. 2022.

[33] A. T. Tibebu, B. Yu, Y. Kassahun, E. Vander Poorten, and P. T. Tran, "Towards autonomous robotic catheter navigation using reinforcement learning," in *Proc. 4th Joint Workshop New Technol. Comput./Robot Assist. Surg.*, 2014, pp. 163–166.

[34] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 880–889, Aug. 2014.

[35] A. Melingui, C. Escande, N. Benoudjit, R. Merzouki, and J. B. Mbede, "Qualitative approach for forward kinematic modeling of a compact bionic handling assistant trunk," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 9353–9358, 2014.

[36] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *Proc. IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2018, pp. 39–45.

[37] S. Satheeshbabu, N. K. Uppalapati, G. Chowdhary, and G. Krishnan, "Open loop position control of soft continuum arm using deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5133–5139.

[38] A. Centurelli, L. Arleo, A. Rizzo, S. Tolu, C. Laschi, and E. Falotico, "Closed-loop dynamic control of a soft manipulator using deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4741–4748, Apr. 2022.

[39] M. Li, R. Kang, D. T. Branson, and J. S. Dai, "Model-free control for continuum robots based on an adaptive Kalman filter," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 286–297, Feb. 2018.

[40] X. Wang et al., "Learning-based visual-strain fusion for eye-in-hand continuum robot pose estimation and control," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 2448–2467, Jun. 2023.

[41] T. George 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, Apr. 2018.

[42] C. D. Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Syst.*, vol. 43, no. 3, pp. 30–65, Jun. 2023.

[43] C. Laschi, T. G. Thuruthel, F. Lida, R. Merzouki, and E. Falotico, "Learning-based control strategies for soft robots: Theory, achievements, and future challenges," *IEEE Control Syst. Mag.*, vol. 43, no. 3, pp. 100–113, Jun. 2023.

[44] B. Zhang and P. Liu, "Model-based and model-free robot control: A review," in *Proc. 8th Int. Conf. Robot Intell. Technol. Appl. (RiTA)*. Singapore: Springer, 2021, pp. 45–55.

[45] C. Schlagenhauf et al., "Control of tendon-driven soft foam robot hands," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2018, pp. 1–7.

[46] G. Runge, M. Wiese, and A. Raatz, "FEM-based training of artificial neural networks for modular soft robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2017, pp. 385–392.

[47] M. Wiese, G. Runge-Borchert, and A. Raatz, "Optimization of neural network hyperparameters for modeling of soft pneumatic actuators," in *New Trends in Medical and Service Robotics: Advances in Theory and Practice*. Cham, Switzerland: Springer, 2019, pp. 199–206.

[48] F. Faure et al., "SOFA: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery* (Studies in Mechanobiology, Tissue Engineering and Biomaterials), vol. 11, Y. Payan, Ed. Berlin, Germany: Springer, Jun. 2012, pp. 283–321. [Online]. Available: https://inria.hal.science/hal-00681539

[49] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2550–2555.

[50] M. Thieffry, A. Kruszewski, C. Duriez, and T.-M. Guerra, "Control design for soft robots based on reduced-order model," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 25–32, Jan. 2019.

[51] M. Gazzola, L. Dudte, A. McCormick, and L. Mahadevan, "Forward and inverse problems in the mechanics of soft filaments," *Roy. Soc. Open Sci.*, vol. 5, no. 6, 2018, Art. no. 171628, doi: 10.1098/rsos.171628.

[52] F. Renda, C. Armanini, V. Lebastard, F. Candelier, and F. Boyer, "A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4006–4013, Jul. 2020.

[53] A. T. Mathew, I. M. B. Hmida, C. Armanini, F. Boyer, and F. Renda, "SoRoSim: A MATLAB toolbox for hybrid rigid–soft robots based on the geometric variable-strain approach," *IEEE Robot. Autom. Mag.*, vol. 30, no. 3, pp. 106–122, Sep. 2022.

[54] H. Jiang et al., "A two-level approach for solving the inverse kinematics of an extensible soft arm considering viscoelastic behavior," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6127–6133.

[55] I. S. Godage, G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Dynamics for variable length multisection continuum arms," *Int. J. Robot. Res.*, vol. 35, no. 6, pp. 695–722, 2016.

[56] E. Milana, F. Stella, B. Gorissen, D. Reynaerts, and C. D. Santina, "Model-based control can improve the performance of artificial cilia," in *Proc. IEEE 4th Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2021, pp. 527–530.

[57] M. H. Namdar Ghalati, H. Ghafarirad, A. A. Suratgar, M. Zareinejad, and M. A. Ahmadi-Pajouh, "Static modeling of soft reinforced bending actuator considering external force constraints," *Soft Robot.*, vol. 9, no. 4, pp. 776–787, Aug. 2022.

[58] R. Grassmann, V. Modes, and J. Burgner-Kahrs, "Learning the forward and inverse kinematics of a 6-DOF concentric tube continuum robot in SE(3)," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5125–5132.

[59] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *Proc. IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2018, pp. 46–53.

[60] Z. Q. Tang, H. L. Heung, K. Y. Tong, and Z. Li, "A novel iterative learning model predictive control method for soft bending actuators," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4004–4010.

[61] Z. Wang and S. Hirai, "Soft gripper dynamics using a line-segment model with an optimization-based parameter identification method," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 624–631, Apr. 2017.

[62] X. Wang et al., "Experimental validation of robot-assisted cardiovascular catheterization: Model-based versus model-free control," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 13, pp. 797–804, 2018.

[63] X. Wang et al., "Eye-in-hand visual servoing enhanced with sparse strain measurement for soft continuum robots," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2161–2168, Apr. 2020.

[64] Y.-Y. Wu and N. Tan, "Model-less feedback control for soft manipulators with Jacobian adaptation," in *Proc. Int. Symp. Auto. Syst. (ISAS)*, Dec. 2020, pp. 217–222.

[65] M. C. Yip, J. A. Sganga, and D. B. Camarillo, "Autonomous control of continuum robot manipulators for complex cardiac ablation tasks," *J. Med. Robot. Res.*, vol. 2, no. 1, 2017, Art. no. 1750002.

[66] M. Verghese, F. Richter, A. Gunn, P. Weissbrod, and M. Yip, "Model-free visual control for continuum robot manipulators via orientation adaptation," in *Proc. Int. Symp. Robot. Res.* Cham, Switzerland: Springer, 2019, pp. 959–970.

[67] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, "Learning closed loop kinematic controllers for continuum manipulators in unstructured environments," *Soft Robot.*, vol. 4, no. 3, pp. 285–296, 2017.

[68] Y. Jin et al., "Model-less feedback control for soft manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2916–2922.

[69] H. Jiang et al., "Hierarchical control of soft manipulators towards unstructured interactions," *Int. J. Robot. Res.*, vol. 40, no. 1, pp. 411–434, 2021.

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

[71] S. Sefati, R. Hegeman, F. Alambeigi, I. Iordachita, and M. Armand, "FBG-based position estimation of highly deformable continuum manipulators: Model-dependent vs. data-driven approaches," in *Proc. Int. Symp. Med. Robot. (ISMR)*, Apr. 2019, pp. 1–6.

[72] W. Xu, J. Chen, H. Y. K. Lau, and H. Ren, "Automate surgical tasks for a flexible serpentine manipulator via learning actuation space trajectory from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4406–4413.

[73] I. M. Loutfi, A. B. Boutchouang, A. Melingui, O. Lakhal, F. B. Motto, and R. Merzouki, "Learning-based approaches for forward kinematic modeling of continuum manipulators," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9899–9904, 2020.

[74] B. Yu, J. d. G. Fernández, and T. Tan, "Probabilistic kinematic model of a robotic catheter for 3D position control," *Soft Robot.*, vol. 6, no. 2, pp. 184–194, 2019.

[75] A. Melingui, J. J. Mvogo Ahanda, O. Lakhal, J. B. Mbede, and R. Merzouki, "Adaptive algorithms for performance improvement of a class of continuum manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1531–1541, Sep. 2018.

[76] G. Fagogenis, C. Bergeles, and P. E. Dupont, "Adaptive nonparametric kinematic modeling of concentric tube robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4324–4329.

[77] Z. Tang, P. Wang, W. Xin, and C. Laschi, "Learning-based approach for a soft assistive robotic arm to achieve simultaneous position and force control," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8315–8322, Jul. 2022.

[78] J. D. Ho et al., "Localized online learning-based control of a soft redundant manipulator under variable loading," *Adv. Robot.*, vol. 32, no. 21, pp. 1168–1183, 2018.

[79] M. Malekzadeh, J. Queißer, and J. J. Steil, "Learning the end-effector pose from demonstration for the bionic handling assistant robot," in *Proc. 9th Int. Workshop Hum.-Friendly Robot.* Genoa, Italy: Springer, 2016, pp. 101–107.

[80] S. Calinon, D. Bruno, M. S. Malekzadeh, T. Nanayakkara, and D. G. Caldwell, "Human–robot skills transfer interfaces for a flexible surgical robot," *Comput. Methods Programs Biomed.*, vol. 116, no. 2, pp. 81–96, 2014.

[81] A. Ataka et al., "Real-time pose estimation and obstacle avoidance for multi-segment continuum manipulator in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 2827–2832.

[82] C. Jang, J. Ha, P. E. Dupont, and F. C. Park, "Toward on-line parameter estimation of concentric tube robots using a mechanics-based kinematic model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 2400–2405.

[83] J. Y. Loo, K. C. Kong, C. P. Tan, and S. G. Nurzaman, "Non-linear system identification and state estimation in a pneumatic based soft continuum robot," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2019, pp. 39–46.

[84] J. Y. Loo, Z. Y. Ding, E. Davies, S. G. Nurzaman, and C. P. Tan, "Curvature and force estimation for a soft finger using an EKF with unknown input optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8506–8512, 2020.

[85] D. C. Rucker and R. J. Webster, "Deflection-based force sensing for continuum robots: A probabilistic approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 3764–3769.

[86] S. Tully, G. Kantor, M. A. Zenati, and H. Choset, "Shape estimation for image-guided surgery with a highly articulated snake robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Sep. 2011, pp. 1353–1358.

[87] C. Kim, S. C. Ryu, and P. E. Dupont, "Real-time adaptive kinematic model estimation of concentric tube robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 3214–3219.

[88] K. Tan, Q. Ji, L. Feng, and M. Törngren, "Edge-enabled adaptive shape estimation of 3D printed soft actuators with Gaussian processes and unscented Kalman filters," *IEEE Trans. Ind. Electron.*, vol. 71, no. 3, pp. 3044–3054, Mar. 2024.

[89] D. Lunni, G. Giordano, E. Sinibaldi, M. Cianchetti, and B. Mazzolai, "Shape estimation based on Kalman filtering: Towards fully soft proprioception," in *Proc. IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2018, pp. 541–546.

[90] Z. Tang et al., "Meta-learning-based optimal control for soft robotic manipulators to interact with unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 982–988.

[91] K. Neumann, M. Rolf, and J. J. Steil, "Reliable integration of continuous constraints into extreme learning machines," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 21, pp. 35–50, Dec. 2013.

[92] M. Giorelli, F. Renda, G. Ferri, and C. Laschi, "A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 5033–5039.

[93] 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. 066003.

[94] G. Soter, A. Conn, H. Hauser, and J. Rossiter, "Bodily aware soft robots: Integration of proprioceptive and exteroceptive sensors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2448–2453.

[95] M. A. Ahmad, C. Gruijthuijsen, M. Ourak, J. Deprest, E. Vander Poorten, and T. Vercauteren, "Shared control of an automatically aligning endoscopic instrument based on convolutional neural networks," in *Proc. 9th Joint Workshop New Technol. Comput./Robot Assist. Surg.*, Genoa, Italy, 2019, pp. 1–2.

[96] W. Xu, J. Chen, H. Y. Lau, and H. Ren, "Data-driven methods towards learning the highly nonlinear inverse kinematics of tendon-driven surgical manipulators," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 13, no. 3, p. e1774, 2017.

[97] R. F. Reinhart and J. J. Steil, "Hybrid mechanical and data-driven modeling improves inverse kinematic control of a soft robot," *Proc. Technol.*, vol. 26, pp. 12–19, Jan. 2016.

[98] R. Reinhart, Z. Shareef, and J. Steil, "Hybrid analytical and data-driven modeling for feed-forward robot control," *Sensors*, vol. 17, no. 2, p. 311, Feb. 2017.

[99] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1270–1277, Dec. 2007.

[100] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural network and Jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 823–834, Aug. 2015.

[101] T. Baaij et al., "Learning 3D shape proprioception for continuum soft robots with multiple magnetic sensors," *Soft Matter*, vol. 19, no. 1, pp. 44–56, 2023.

[102] P. Hyatt and M. D. Killpack, "Real-time nonlinear model predictive control of robots using a graphics processing unit," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1468–1475, Apr. 2020.

[103] G. Fang, Y. Tian, Z.-X. Yang, J. M. P. Geraedts, and C. C. L. Wang, "Efficient jacobian-based inverse kinematics with Sim-to-Real transfer of soft robots by learning," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 6, pp. 5296–5306, Dec. 2022.

[104] A. Kuntz, A. Sethi, R. J. Webster, and R. Alterovitz, "Learning the complete shape of concentric tube robots," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 2, pp. 140–147, May 2020.

[105] R. Grassmann and J. Burgner-Kahrs, "On the merits of joint space and orientation representations in learning the forward kinematics in SE(3)," in *Proc. Robot., Sci. Syst. Conf.*, 2019, pp. 1–10. [Online]. Available: http://www.roboticsproceedings.org/rss15/p17.pdf

[106] J. Liu, P. Borja, and C. D. Santina, "Physics-informed neural networks to model and control robots: A theoretical and experimental investigation," 2023, *arXiv:2305.05375*.

[107] F. Pique, H. T. Kalidindi, L. Fruzzetti, C. Laschi, A. Menciassi, and E. Falotico, "Controlling soft robotic arms using continual learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5469–5476, Apr. 2022.

[108] A. Melingui, O. Lakhal, B. Daachi, J. B. Mbede, and R. Merzouki, "Adaptive neural network control of a compact bionic handling arm," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 6, pp. 2862–2875, Dec. 2015.

[109] X. Li et al., "Deep learning for haptic feedback of flexible endoscopic robot without prior knowledge on sheath configuration," *Int. J. Mech. Sci.*, vol. 163, Nov. 2019, Art. no. 105129.

[110] S. Yao, R. Tang, L. Bai, H. Yan, H. Ren, and L. Liu, "An RNN-LSTM enhanced compact and affordable micro force sensing system for interventional continuum robots with interchangeable end-effector instruments," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 4008711.

[111] 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. Autom. Lett.*, vol. 6, no. 2, pp. 951–957, Apr. 2021.

[112] 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.

[113] Z. Zhou et al., "A sensory soft robotic gripper capable of learning-based object recognition and force-controlled grasping," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 1, pp. 844–854, Jan. 2024.

[114] Z. Chen, X. Ren, M. Bernabei, V. Mainardi, G. Ciuti, and C. Stefanini, "A hybrid adaptive controller for soft robot interchangeability," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 875–882, Jan. 2024.

[115] E. Almanzor, F. Ye, J. Shi, T. G. Thuruthel, H. A. Wurdemann, and F. Iida, "Static shape control of soft continuum robots using deep visual inverse kinematic models," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 2973–2988, Aug. 2023.

[116] N. Liang, R. M. Grassmann, S. Lilge, and J. Burgner-Kahrs, "Learning-based inverse kinematics from shape as input for concentric tube continuum robots," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2021, pp. 1387–1393.

[117] U. Yoo, H. Zhao, A. Altamirano, W. Yuan, and C. Feng, "Toward zero-shot sim-to-real transfer learning for pneumatic soft robot 3D proprioceptive sensing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 544–551.

[118] Y. Zhang, J. Gao, H. Yang, and L. Hao, "A novel hysteresis modelling method with improved generalization capability for pneumatic artificial muscles," *Smart Mater. Struct.*, vol. 28, no. 10, Oct. 2019, Art. no. 105014.

[119] Y. Chen et al., "Safety-enhanced motion planning for flexible surgical manipulator using neural dynamics," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1711–1723, Sep. 2017.

[120] Z. Chen, M. Bernabei, V. Mainardi, X. Ren, G. Ciuti, and C. Stefanini, "A novel and accurate BiLSTM configuration controller for modular soft robots with module number adaptability," 2024, *arXiv:2401.10997*.

[121] S. Sapai, J. Y. Loo, Z. Y. Ding, C. P. Tan, V. M. Baskaran, and S. G. Nurzaman, "A deep learning framework for soft robots with synthetic data," *Soft Robot.*, vol. 10, no. 6, pp. 1224–1240, Dec. 2023.

[122] A. Melingui, R. Merzouki, J. B. Mbede, C. Escande, and N. Benoudjit, "Neural networks based approach for inverse kinematic modeling of a compact bionic handling assistant trunk," in *Proc. IEEE 23rd Int. Symp. Ind. Electron. (ISIE)*, Jun. 2014, pp. 1239–1244.

[123] Y. Engel, P. Szabo, and D. Volkinshtein, "Learning to control an octopus arm with Gaussian process temporal difference methods," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18, 2005, pp. 1–8.

[124] Y. Kassahun, B. Yu, and E. Vander Poorten, "Learning catheter-aorta interaction model using joint probability densities," in *Proc. 3rd Joint Workshop New Technol. Comput./Robot Assist. Surg.*, 2013, pp. 158–160.

[125] J. Chen, H. Y. K. Lau, W. Xu, and H. Ren, "Towards transferring skills to flexible surgical robots with programming by demonstration and reinforcement learning," in *Proc. 8th Int. Conf. Adv. Comput. Intell. (ICACI)*, Feb. 2016, pp. 378–384.

[126] X. Dong, J. Zhang, L. Cheng, W. Xu, H. Su, and T. Mei, "A policy gradient algorithm integrating long and short-term rewards for soft continuum arm control," *Sci. China Technol. Sci.*, vol. 65, no. 10, pp. 2409–2419, Oct. 2022.

[127] G. Ji et al., "Towards safe control of continuum manipulator using shielded multiagent reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7461–7468, Oct. 2021.

[128] X. Liu, R. Gasoto, Z. Jiang, C. Onal, and J. Fu, "Learning to locomote with artificial neural-network and CPG-based control in a soft snake robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 7758–7765.

[129] X. Liu, C. Onal, and J. Fu, "Learning contact-aware CPG-based locomotion in a soft snake robot," 2021, *arXiv:2105.04608*.

[130] Y. Lu et al., "Autonomous intelligent navigation for flexible endoscopy using monocular depth guidance and 3-D shape planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 1–7.

[131] Y. Gan et al., "A reinforcement learning method for motion control with constraints on an HPN arm," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12006–12013, Oct. 2022.

[132] O. Mumini Omisore, T. Akinyemi, W. Duan, W. Du, and L. Wang, "A novel sample-efficient deep reinforcement learning with episodic policy transfer for PID-based control in cardiac catheterization robots," 2021, *arXiv:2110.14941*.

[133] Y. Li, X. Wang, and K.-W. Kwok, "Towards adaptive continuous control of soft robotic manipulator using reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 7074–7081.

[134] Q. Wu et al., "Position control of cable-driven robotic soft arm based on deep reinforcement learning," *Information*, vol. 11, no. 6, p. 310, 2020.

[135] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[136] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[137] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[138] M. S. Nazeer, D. Bianchi, G. Campinoti, C. Laschi, and E. Falotico, "Policy adaptation using an online regressing network in a soft robotic arm," in *Proc. IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2023, pp. 1–7.

Zixi Chen (Graduate Student Member, IEEE) received the M.Sc. degree in control systems from Imperial College in 2021. He is currently pursuing the Ph.D. degree in biorobotics from Scuola Superiore Sant'Anna of Pisa. His research interest includes optical tactile sensors and soft robot control with neural networks.

Federico Renda (Member, IEEE) received the B.Sc. and M.Sc. degrees in biomedical engineering from the University of Pisa, Pisa, Italy, in 2007 and 2009, respectively, and the Ph.D. degree in biorobotics from The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, in 2014.

He is currently an Associate Professor with the Department of Mechanical Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His research interests include dynamic modeling and control of soft and continuum robots using principles of geometric mechanics.

Alexia Le Gall (Student Member, IEEE) received the Engineering degree in mechanical conception and the M.Sc. degree in mechatronics from the University of Technology of Compiègne, France, in 2022. She is currently pursuing the Ph.D. degree in biorobotics from Scuola Superiore Sant'Anna, with a specialization in soft robotic.

Her research interests include mechanical design and soft robotic actuation and fabrication methods.

Lorenzo Mocellin received the B.Sc. degree in biomedical engineering from the University of Padua in 2019 and the M.Sc. degree in bioengineering from the University of Trieste in 2022, with a thesis on cardiovascular fluid mechanics modeling. He is currently pursuing the Ph.D. degree with The BioRobotics Institute, Scuola Superiore Sant'Anna.

He works with the Surgical Robotics Laboratory on the design and development of surgical robotic tools. His research interests include miniaturized medical devices, surgical robotic end-effectors, origami-inspired structures, compliant mechanisms, and biomaterials.

Matteo Bernabei received the bachelor's degree in industrial engineering and the master's degree in biomedical engineering from the Campus Bio-Medico University of Rome, Italy, in 2020 and 2022, respectively. He is currently pursuing the Ph.D. degree in biorobotics with Scuola Superiore Sant'Anna (SSSA), Italy. In 2021, as part of his formation, he spent six months as a Research Intern with Boston Children's Hospital to work on advanced methods for analysis of EEG signals for the optimization of pre-surgery planning in children with drug-resistant epilepsy. His current research interests include endoluminal and minimally invasive surgical robotics.

Théo Dangel received the M.Eng. degree in mechanical engineering from SUPMICROTECH-ENSMM, Besançon, France, and the M.S. degree in electronics from Tokyo Denki University, Tokyo, Japan, in 2022. He is currently pursuing the Ph.D. degree in biorobotics with Scuola Superiore Sant'Anna, Pisa, Italy. His research interests include actuators for soft robotic.

Gastone Ciuti (Senior Member, IEEE) received the master's degree (Hons.) in biomedical engineering from the University of Pisa, Pisa, Italy, in 2008, and the Ph.D. degree (Hons.) in biorobotics from The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, in 2011. He has been a Visiting Professor with Sorbonne University, Paris, France, and Beijing Institute of Technology, Beijing, China; and a Visiting Student with Vanderbilt University, Nashville, TN, USA, and the Imperial College London, London, U.K. He is currently an Associate Professor of Bioengineering with Scuola Superiore Sant'Anna, leading the Healthcare Mechatronics Laboratory. He is the coauthor of more than 110 international peer-reviewed articles on medical robotics and the inventor of more than 15 patents. His research interests include robot/computer-assisted platforms, such as teleoperated and autonomous magnetic-based robotic platforms for navigation, localization, and tracking of smart and innovative devices in guided and targeted minimally invasive surgical and diagnostic applications, e.g., advanced capsule endoscopy. He is a member of the Technical Committee in biorobotics of the IEEE Engineering in Medicine and Biology Society (EMBS). He is an Associate Editor of IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, and IEEE TRANSACTIONS ON MEDICAL ROBOTICS AND BIONICS.

Matteo Cianchetti (Member, IEEE) received the M.Sc. degree (cum laude) in biomedical engineering from the University of Pisa, Italy, in 2007, and the Ph.D. degree in biorobotics from Scuola Superiore Sant'Anna. He is currently an Assistant Professor with The BioRobotics Institute, Scuola Superiore Sant'Anna, leading the Soft Mechatronics for Biorobotics Laboratory. He is the author or coauthor of more than 100 international peer-reviewed articles. He has been involved in EU-funded projects with a main focus on the development of soft robotics technologies. His main research interests include bioinspired robotics and the study and development of new systems and technologies based on soft/flexible materials for soft actuators, smart compliant sensors, and flexible mechanisms. He is the Chair of the IEEE TC on Soft Robotics. He regularly serves as a reviewer for more than ten international journals.

Cesare Stefanini (Member, IEEE) received the M.Sc. degree (Hons.) in mechanical engineering and the Ph.D. degree (Hons.) in microengineering from Scuola Superiore Sant'Anna (SSSA), Pisa, Italy, in 1997 and 2002, respectively.

He has been a Visiting Researcher with the Center for Design Research, Stanford University, and the Director of the Healthcare Engineering Innovation Center, Khalifa University, Abu Dhabi, United Arab Emirates. He is currently a Professor and the Director of The BioRobotics Institute, SSSA, where he is also the Head of the Creative Engineering Laboratory. He is the author or coauthor of more than 200 articles in refereed international journals and on international conference proceedings. He is the inventor of 15 international patents, nine of which are industrially exploited by international companies. His research activity is applied to different fields, including underwater robotics, bioinspired systems, biomechatronics, and micromechatronics for medical and industrial applications.

Prof. Stefanini is a member of the Academy of Scientists of United Arab Emirates and the IEEE Societies Robotics and Automation (RAS) and Engineering in Medicine and Biology (EMBS). He was a recipient of the "Intuitive Surgical Research Award." He received international recognition for the development of novel actuators for microrobots.

Open Access funding provided by 'Scuola Superiore "S.Anna" di Studi Universitari e di Perfezionamento' within the CRUI CARE Agreement