2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# Simulation-Augmented Hysteresis Compensation in Continuum Surgical Robots via Residual Learning

Haolin Jiang, Thiusius R. Savarimuthu, Di Wu*

*Abstract*— **Continuum robots show promise for minimally invasive surgery, but suffer from tip positioning errors due to hysteresis. Pure analytical models struggle to capture nonlinear effects, while data-driven methods typically demand extensive amounts of robot training data, potentially leading to increased system wear. To overcome the limitations of both analytical and data-driven methods, we propose a novel Real2Sim2Real framework that establishes a Finite-Element (FE) digital twin of a continuum robot. The framework is structured in two stages. We first collect a small amount of real hysteresis data and identify the tendon-space residual actuation displacement required to minimize the pose error between the finite-element (FE) model and the physical robot. These residual displacements are then applied as auxiliary tendon inputs to enhance the realism of the simulation, ensuring that the hysteresis behavior of the digital twin aligns with that of the physical system. This enables the generation of a large-scale, high-fidelity synthetic dataset at minimal cost, substantially expanding the available training data. In the second stage, we train a deep-learning (DL)–based inverse kinematics model that maps desired tip poses to primary tendon displacements, using a hybrid dataset composed of both real and simulated data. The key advantage of this approach is that it achieves a high-accuracy controller while requiring only a small amount of real robot data. The trained controller was evaluated on various 2D trajectories. Results demonstrate effective hysteresis compensation, achieving tip positioning errors between 1.344 mm and 3.678 mm on unseen trajectories, even under large bending conditions. When normalized by the trajectory range, the error falls between 2.0% and 5.3%. Compared to a nominal model-based controller, the proposed simulation-augmented inverse controller achieves substantially higher precision and superior hysteresis compensation in continuum surgical robots.**

## I. INTRODUCTION

Continuum robots have become key enablers for minimally invasive interventions due to their flexibility and dexterity [1]. These robots can navigate complex anatomical paths and access hard-to-reach regions of the human body. However, achieving precise control remains challenging, with one of the reasons being unmodeled nonlinear effects such as hysteresis [2]. Friction between tendons and the sheath, elasticity of the cables and robot body materials, and tendon slack can lead to asymmetric loading–unloading behavior, making the robot's configuration path-dependent [3], [4]. This has significant implications for interventional procedures. For example, in minimally invasive procedures such as endovascular interventions, hysteresis can cause delayed responses at the catheter tip, increasing the risk of tissue damage when navigating constrained anatomical pathways. Therefore, hysteresis compensation is critical for improving safety margins and procedural reliability, particularly because integrating tip-mounted sensors remains clinically challenging. Traditional analytical models (e.g., Prandtl–Ishlinskii model, Preisach model) can capture hysteresis behavior and offer good physical interpretability, but they are often less accurate than data-driven methods, as demonstrated in [5]. On the other hand, data-driven approaches have shown promise in modeling or compensating for these effects. For example, Long Short-Term Memory (LSTM) networks have been used to model catheter tip hysteresis [6], [3], [5], and Gaussian mixture models have improved kinematic accuracy by 60–70% via hysteresis compensation in tendon-driven surgical robots [7]. However, a purely learning-based strategy requires extensive real data collected from the physical system, which can cause system wear or degradation and is time-consuming.

To address the above challenges, we present a *Real2Sim2Real* hysteresis compensation method that leverages a high-fidelity Finite-Element (FE) simulation based on the Simulation Open Framework Architecture (SOFA) [8], combined with deep learning, to compensate for hysteresis and establish an inverse kinematics model for controlling a tendon-driven continuum surgical robot (see Fig. 1). High-resolution finite-element (FE) modeling captures continuum robot compliance and deformation under tendon pulls [9], [8], [10], providing a physics-rich basis for kinematics. However, purely physics-based simulators exhibit sim-to-real gaps due to unmodeled hysteretic effects such as friction and material creep [4], [2]. To address this, we introduce residual actuation correction. In the first stage, a Real-to-Sim calibration is performed: tip poses are measured on the real robot, and a quadratic programming (QP) approach is employed to compute artificial *auxiliary* actuator displacements within the simulation, ensuring that the simulated poses align with the measured ones. This digital twin inspired calibration yields residual correction profiles that represent unmodeled hysteresis, enabling the FE simulator to closely mimic real robot behavior. In the second stage, we train a Sim-to-Real learning-based inverse kinematics controller. We acquire a hybrid dataset composed of real robot data and synthetic data generated by the calibrated simulator. A neural network is trained to map a desired distal tip pose to the primary tendon displacements.

In summary, the contributions of this work are three-fold: (1) the establishment of a two-stage Real2Sim2Real

H. Jiang, T. R. Savarimuth, and D. Wu are with the Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, 5230 Odense, Denmark. D. Wu is also with the Danish Institute for Advanced Study (DIAS), 5230 Odense, Denmark. *(*Corresponding author: Di Wu, email: diwu@mmmi.sdu.dk)*

framework for continuum surgical robots, featuring a high-fidelity finite-element (FE) simulator that is calibrated with sparse real data to closely resemble the physical robot and subsequently generate synthetic data; (2) the development of a learning-based controller that leverages both synthetic and sparse real data to compensate for hysteresis and achieves accurate open-loop tip control in 2D space; and (3) an experimental validation demonstrating enhanced control accuracy and effective hysteresis compensation in 2D trajectory tracking, which is crucial for precise tip positioning in minimally invasive interventions.

## II. BACKGROUND AND RELATED WORK

### A. Hysteresis in continuum robots

Continuum manipulators inherently exhibit hysteresis due to tendon routing, material and elasticity. As a result, the kinematic model can depend on the path taken (load history) rather than just the current inputs [5], [11]. In general, hysteresis compensation in continuum robots can be categorized into two main approaches: analytical modeling [12] and learning–based methods [13].

### B. Finite-element simulation and digital twins

High-fidelity physics models are important for understanding and controlling soft/continuum robots. Finite-element methods (FEM) have been widely adopted to model continuum robot backbones and tendons, enabling accurate representation of compliance and large deformations [9], [8]. Software frameworks like SOFA provide toolkits for building such *digital twins* of soft robots, which can simulate deformations under applied tendon forces in real time [10], [9], [14]. Real-time FEM has also been used directly for feedback control of elastic soft robots [15]. These FE-based simulations serve as valuable proxies of the real robot for design optimization, control strategy testing, and even operator training. However, purely physics-based digital twins often struggle to capture non-linear phenomena like tendon friction, slack, and material hysteresis, leading to a sim-to-real mismatch. Prior work reports that even a fine-resolution Cosserat or FEM model can diverge from reality if cable frictional forces are not modeled [4]. Thus, researchers have begun augmenting simulations with empirical data. Ferrentino *et al.* recently demonstrated FE simulation of a soft actuator in SOFA and highlighted the need for parameter calibration to align the model with physical tests [8]. In our work, a calibrated digital twin serves as the foundation for synthetic data generation. We enhance the twin with learned residuals, combining the high accuracy of FE modeling with the good adaptability of data-driven calibration.

### C. Learning-based inverse kinematic modeling and sim-to-real

Data-driven control of continuum robots has become popular, as their kinematics are difficult to model due to nonlinear effects. One paradigm is to directly learn the inverse (differential) kinematics from data, bypassing the need for an exact model. Yip *et al.* introduced a model-less adaptive controller for continuum manipulators in constrained environments, demonstrating robust performance without an explicit model [16]. Wu *et al.* trained a LSTM-based controller for a robotic catheter, achieving precise tip tracking despite severe rate-dependent hysteresis [5], [17]. Reinforcement learning (RL) has also been explored. Ji *et al.* proposed a shielded multi-agent RL scheme for safe continuum manipulator control [18], and Nazeer *et al.* applied RL-based adaptive control to a soft arm to attain high-precision reaching tasks [19]. A major challenge for learned controllers is transferring policies from simulation to the real robot (sim-to-real), as simulation inaccuracies can cause controllers to fail on hardware. Techniques such as domain randomization and augmentation of training data have been used to bridge this gap [20], [11]. Yao *et al.* demonstrated that training a guidewire navigation policy entirely in a stochastic simulation (with randomized physical parameters) allowed successful deployment in a real vascular robot [20]. Similarly, Cho *et al.* trained a deep policy for autonomous guidewire manipulation in simulation and achieved effective sim-to-real transfer using behavior cloning to fine-tune on real data [11]. Gao *et al.* introduced a novel residual learning approach based on a differentiable simulator [21]. These efforts highlight the benefit of combining simulation environments with learning-based methods: simulators provide a generalizable foundation, while learning handles complex residual effects. Our work follows this philosophy by applying learned residual corrections to a physics-based model and training an inverse kinematics network with both simulated and real data.

## III. PURPOSE-BUILT CONTINUUM ROBOTIC SYSTEM

A continuum robotic platform was built for data collection and validation. The robot has two degrees of freedom (2-DOF) in bending, actuated by four tendons attached to the distal end at orthogonal corners. The platform (Fig. 1b) employs four servo motors (Dynamixel XL430, Robotis Co., Ltd., Korea) operating in position-control mode, where encoder ticks correspond to commanded tendon displacements.

The distal segment is 3D printed in thermoplastic polyurethane (TPU) to provide body flexibility, with a length of 80 mm and an outer diameter of 12 mm. A central channel (diameter 6 mm) houses a 6-DOF Electromagnetic (EM) sensor (trackSTAR, NDI, Canada) fixed at the tip, while four peripheral channels (diameter 1.5 mm) are integrated into the wall to guide the tendons.

## IV. METHOD

We investigate a tendon-driven continuum robot equipped with four tendons along its backbone. Our ultimate objective is to develop an open-loop inverse controller that compensates for hysteresis without relying on real-time feedback. Note that this paper focuses on hysteresis in the bending direction of continuum robots. As a foundational step, we calibrate a high-fidelity digital twin in a finite-element simulator to accurately replicate the real robot's distal tip trajectory under the same tendon commands, even in the presence of unmodeled hysteretic effects. We achieve this in two stages:

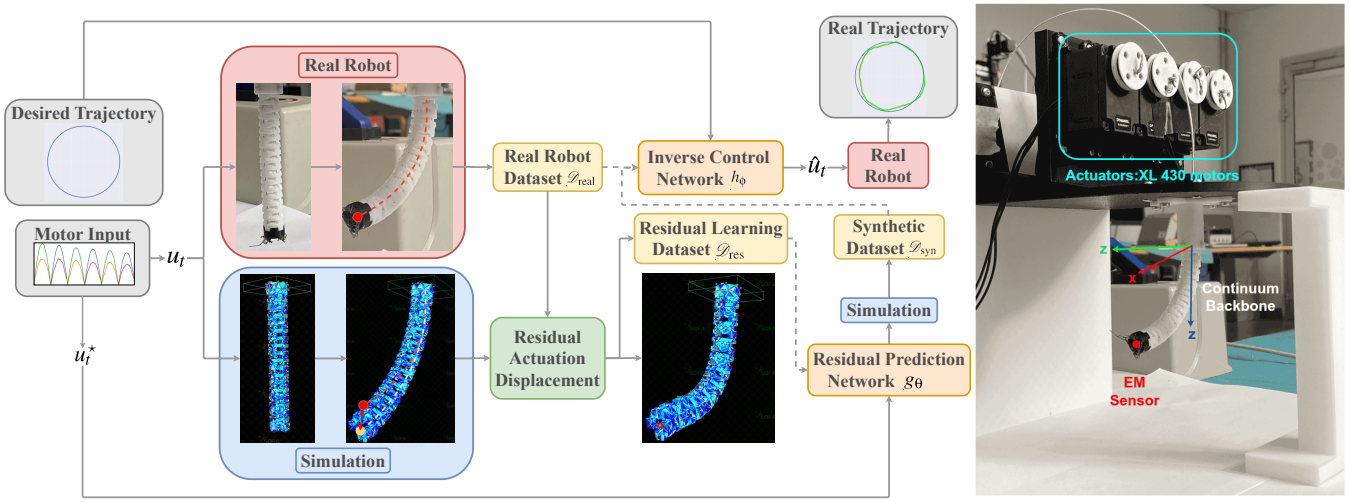


**Fig. 1: System overview and experimental platform. (a)** Real2Sim2Real pipeline for hysteresis-aware control of a tendon-driven continuum robot. A calibrated SOFA-based digital twin [10], [9] is augmented with a learned forward hysteresis model—*Compensatory Residual Actuation* (CoReAct)—to reproduce hysteretic effects observed on hardware. The twin then synthesizes safe, coverage-expanding data to train a controller. Dashed lines indicate the data flow used for training the neural network, while solid lines represent the actual input–output relationships on the real robot/simulation environment. The proposed *Equality-Driven Inverse Compensation Control* (EDIC) computes actuation sequences that track a desired tip trajectory despite hysteresis and model mismatch. **(b)** A purpose-built continuum robotic system for data collection and validation: Four servo motors drive four tendons attached to the distal tip, enabling controlled bending. The robot body is made of TPU, and an EM sensor is inserted through the working channel to measure the 6-DOF tip pose.

(i) a *forward* stage that models the sim–real mismatch as a low-dimensional *tendon-space residual actuation* applied via auxiliary tendons, and (ii) an *inverse* stage that uses a hybrid synthetic augmented dataset to learn a mapping from desired poses to primary tendon commands.

### *A. Simulation preliminaries: tendons and actuator-space projection*

**Tendons in the simulator.** In the FE twin, each physical tendon is modeled as a *displacement-controlled cable constraint* routed along a polyline embedded in the backbone mesh. At each quasi-static step, it applies an equivalent retraction stroke and induces a unilateral axial tension. We refer to the four tendons that exist on both the physical robot and the simulator as *primary* tendons. In addition, we introduce *auxiliary* tendons, which are virtual and exist only in the simulator to account for residual corrections [9].

**Simulator I/O notation.** Let $u_t \in \mathbb{R}^4$ be the primary tendon strokes at step $t$, let $s_t \in \mathbb{R}^{n_s}$ be the auxiliary (residual) tendon strokes enforced as equality constraints, and let $y_t^{\text{sim}} \in \mathbb{R}^3$ denote the distal-end pose extracted from the converged FE state. One quasi-static Newton+QP solve defines the step mapping

$$(x_t, y_t^{\text{sim}}) = \mathscr{F}(x_{t-1}, u_t, s_t), \qquad y_t^{\text{sim}} = \mathscr{S}(u_t, s_t), \quad (1)$$

where $x_t \in \mathbb{R}^{3n}$ are all nodal positions at convergence. When only the primary tendons are active, we write $y_t^{\text{sim}} = \mathscr{S}(u_t, 0)$.

**Actuator-space projection.** Let $x \in \mathbb{R}^{3n}$ be the stacked nodal positions of a tetrahedral FE model, and let $K(x) \in \mathbb{R}^{3n\times 3n}$ be its tangent stiffness matrix at a Newton iterate. Let $J(x) \in \mathbb{R}^{m\times 3n}$ collect the linearized constraint Jacobian rows (including effectors and tendons). Eliminating the nodal increments $dx$ yields the standard projection into the constraint space [9], [15]:

$$\begin{aligned} \delta &= \delta_{\text{free}} + W\lambda, \\ W &:= JK^{-1}J^\top, \\ \delta_{\text{free}} &:= JK^{-1}\big(p - f_{\text{int}}\big), \end{aligned} \quad (2)$$

where $\delta \in \mathbb{R}^m$ are generalized displacements and $\lambda \in \mathbb{R}^m$ are the associated Lagrange multipliers (generalized forces). $W$ is the (symmetric positive-semidefinite) *Delassus operator*. For a cable constraint $j$, $\delta_j$ is the (linearized) equivalent stroke and $\lambda_j$ is its tension.

We partition the constraint rows as

$$\delta = \begin{bmatrix} \delta_e \\ \delta_a \\ \delta_s \end{bmatrix}, \qquad \lambda = \begin{bmatrix} \lambda_e \\ \lambda_a \\ \lambda_s \end{bmatrix},$$

where $e$ indexes the task-space effectors at the distal end, $a$ indexes the four *primary* tendons, and $s$ indexes the *auxiliary* tendons. (In the FE model, each tendon is implemented as a displacement-controlled cable that applies a specified stroke and develops tension along the backbone.) Block-partitioning (2) gives

$$\begin{bmatrix} \delta_e \\ \delta_a \\ \delta_s \end{bmatrix} = \begin{bmatrix} \delta_e^{\text{free}} \\ \delta_a^{\text{free}} \\ \delta_s^{\text{free}} \end{bmatrix} + \begin{bmatrix} W_{ee} & W_{ea} & W_{es} \\ W_{ae} & W_{aa} & W_{as} \\ W_{se} & W_{sa} & W_{ss} \end{bmatrix} \begin{bmatrix} \lambda_e \\ \lambda_a \\ \lambda_s \end{bmatrix}. \quad (3)$$

We control the primary tendons by prescribing per-step strokes $\delta_a = u$ (known inputs) as equality constraints, while

the effector rows and auxiliary tendons participate in a quadratic program (QP) solve.

**QP solve per Newton step.** [9], [15] Given an effector goal encoded as $\delta_e = 0$ and known primary strokes $\delta_a = u$, we solve

$$\begin{aligned} \min_{\lambda} \quad & \tfrac{1}{2}\|\delta_e\|_2^2 + \tfrac{1}{2}\lambda^\top R\lambda \\ \text{s.t.} \quad & \delta = \delta_{\text{free}} + W\lambda, \qquad \delta_a = u, \qquad |\lambda| \le \bar{\lambda}, \end{aligned} \tag{4}$$

with a small $R \succeq 0$ for numerical uniqueness and box constraints (here $\bar{\lambda} = \lambda_{\max}\mathbf{1}$). After this QP, the achieved generalized displacements are obtained from Equation (2).

### *B. Forward hysteresis modeling via* Compensatory Residual Actuation (CoReAct)

*a) Problem formulation:* Let $u_{1:T}$ be a sequence of primary-tendon displacements. The simulator's distal trajectory under *only* primary actuation is $\{y_t^{\text{sim}}\}_{t=1}^T$, whereas the real robot (with friction, anisotropy, and hysteresis) produces $\{y_t^{\text{real}}\}_{t=1}^T$.

**Physics intuition.** Under quasi-static actuation with isotropic bending stiffness and frictionless tendons, a sinusoidal primary command with a 90° phase offset between orthogonal cables yields a circular distal tip path in simulation. On hardware, three effects distort this ideal motion: (i) *anisotropy* ($b_x \neq b_y$) biases bending toward principal axes, (ii) *tendon friction* causes direction-dependent tension losses (a capstan-like effect), and (iii) *rate-independent hysteresis* adds a lag that depends on the input history. A compact input–output description capturing these effects is:

$$\begin{aligned} y_t^{\text{sim}} &= \mathscr{S}(u_t, 0) \quad \text{(ideal FE twin)}, \\ y_t^{\text{real}} &= \mathscr{S}(u_t, 0) + \mathscr{M}(u_t, \dot{u}_t, \xi_t) + \mathscr{H}[u]_t, \\ \xi_{t+1} &= \Psi(\xi_t, u_t), \end{aligned} \tag{5}$$

where $\mathscr{M}$ lumps the frictional and anisotropic effects and $\mathscr{H}$ represents hysteresis. Consequently, a commanded circular trajectory in simulation appears as a *diamond* on the real robot (see Figure 2).

**Mathematical mismatch model.** Define the pose error $e_t := y_t^{\text{real}} - y_t^{\text{sim}}$. A first-order local approximation along a trajectory $(u_t, y_t^{\text{sim}})$ is

$$e_t \approx A_t\, e_{t-1} + B_t\, u_t + C_t\, \dot{u}_t + d_t, \tag{6}$$

where $A_t, B_t, C_t, d_t$ capture the local influence of anisotropy, friction, and hysteresis (as shown in Equation (5)). Rather than explicitly identifying these parameters, we directly cancel $e_t$ by injecting auxiliary inputs via $\delta_s = s_t$ (through the Delassus operator (3)). Because $\delta_e$ depends affinely on $s_t$, we can choose $s_t$ to minimize the residual $r_t := y_t^{\text{real}} - y_t^{\text{sim}}$ via a Tikhonov-regularized least-squares solve. This yields a closed-form solution $s_t^\star$ that drives $\delta_e \approx 0$. Notably, this $s_t^\star$ is precisely the reduced solution achieved by the physics-based QP in Equation (7). When the auxiliary subspace can fully actuate the effectors ($W_{es}$ full row rank), the residual $\|\delta_e\|$ can be driven arbitrarily close to zero as $\gamma \to 0$.

In practice, this calibration allows collecting labeled pose–input pairs beyond the originally observed workspace

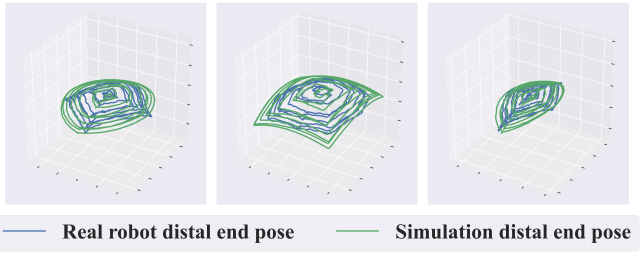


**Fig. 2: Real2Sim gap under identical inputs.** For three command patterns (**Left**: circle, **Middle**: square, **Right**: ellipse), the calibrated simulator reproduces the intended shapes, whereas the physical robot consistently yields a diamond-like trajectory—revealing unmodeled hysteresis and anisotropy.

(e.g., $|d_x|$ or $|d_y| > 25\,\text{mm}$ in our setup). The equality-field actuation stabilizes the digital twin, yielding valid $(u_t, y_t)$ pairs outside the real-data range while preserving physical plausibility.

*b) Key idea:* ***residual actuation*** **as equality rows** We introduce $n_s$ virtual auxiliary tendons along the backbone and *define* a residual generalized displacement $s_t \in \mathbb{R}^{n_s}$ such that when $s_t$ is enforced as an equality in the simulator, the distal effector exactly matches the *measured* real pose at step $t$. Concretely, at each step $t$ we solve the physics-regularized QP [22], [9]:

$$\begin{aligned} \min_{\lambda_e, \lambda_a, \lambda_s} \quad & \tfrac{1}{2}\|\delta_e\|_2^2 + \tfrac{1}{2}\lambda^\top R\lambda \\ \text{s.t.} \quad & \delta = \delta_{\text{free}} + W\lambda, \qquad \delta_a = u_t, \\ & \delta_e = y_t^{\text{goal}} - y_t^{\text{sim}}, \qquad \delta_s = s_t, \end{aligned} \tag{7}$$

where $y_t^{\text{goal}} := y_t^{\text{real}}$ serves as the effector target. We then choose $s_t$ that minimizes $\|\delta_e\|^2$; equivalently, we solve the reduced least-squares in $s$:

$$s_t^\star := \arg\min_s \tfrac{1}{2}\left\| \delta_e^{\text{free}} + W_{ea}\lambda_a(u_t) + W_{es}\lambda_s(s) \right\|_2^2 + \tfrac{\gamma}{2}\|s\|_2^2, \tag{8}$$

where $\lambda_a(u_t)$ and $\lambda_s(s)$ are obtained from the equality constraints $\delta_a = u_t$ and $\delta_s = s$ via the block system (3). The Tikhonov term $\gamma\|s\|_2^2$ regularizes the solution if some auxiliary directions are redundant. Assuming $W_{es}$ has full row rank (auxiliary inputs genuinely affect all effector DOFs), the minimizer $s_t^\star$ is unique.

*c) Data for residual learning:* Applying Equation (7) over all $T$ steps yields a dataset $\mathscr{D}_{\text{res}} = \left\{ \left(u_t, y_t^{\text{real}}, s_t^\star\right) \right\}_{t=1}^T$, which we use to train a neural network:

$$g_\theta : \ (u_{t-w:t},\, y_{t-w:t}) \mapsto \hat{s}_t, \qquad y_t := y_t^{\text{real}}, \tag{9}$$

by minimizing

$$\mathscr{L}_{\text{res}}(\theta) = \sum_{\left((u_t, y_t), s_t^\star\right) \in \mathscr{D}_{\text{res}}} \left\| g_\theta(u_{t-w:t}, y_{t-w:t}) - s_t^\star \right\|_2^2 + \beta_g \|\theta\|_2^2. \tag{10}$$

Here, $w$ is the window size, and we use a window of inputs $u_{t-w:t}$ to account for the history-dependent hysteresis effect in the system.

At deployment, the real pose is unavailable, so $g_\theta$ operates *in*

*closed-loop* using the simulator's own tip feedback. Starting from an initial $\tilde{y}_0$, for each $t = 1, \ldots, T$:

$$\hat{s}_t = g_\theta(u_{t-w:t}, \tilde{y}_{t-w-1:t-1}), \qquad (x_t, \tilde{y}_t) = \mathscr{F}(x_{t-1}, u_t, \hat{s}_t). \tag{11}$$

### C. Equality-Driven Inverse Compensation Control (EDIC)

*a) Synthetic data generation via in-sim closed loop:* CoReAct effectively turns the calibrated twin into a high-fidelity data generator: for any novel primary sequence $u^\star_{1:T'}$, we roll out the simulator while injecting $\delta_s = \hat{s}_t = g_\theta(u^\star_{t-w:t}, \tilde{y}_{t-w-1:t-1})$ as an auxiliary equality each step, producing a simulated trajectory $\tilde{y}_{1:T'}$ that closely matches the real robot. We thus construct a hybrid dataset by combining (i) $N_r$ real trajectories and (ii) $N_s$ synthetic trajectories from such CoReAct rollouts, thereby extending coverage beyond the original real-data regime: $\mathscr{D}_{\text{real}} = \{(u_t, y_t^{\text{real}})\}_{t=1}^{T}$, $\mathscr{D}_{\text{syn}} = \{(u^\star_t, \tilde{y}_t)\}_{t=1}^{T'}$ from Equation (11), $\mathscr{D}_{\text{hyb}} = \mathscr{D}_{\text{real}} \cup \mathscr{D}_{\text{syn}}$.

*b) Inverse control network:* We train a neural network [17] $h_\phi$ to compensate for friction, backlash, and hysteresis and map a desired distal end position to the four primary tendon strokes:

$$h_\phi : \mathbb{R}^{w\times 2} \to \mathbb{R}^4, \qquad \hat{u}_t = h_\phi(\mathbf{y}_{t-w:t}), \tag{12}$$

with supervised loss

$$\mathscr{L}_{\text{sup}}(\phi) = \sum_{(u_t, y_t)\in\mathscr{D}_{\text{hyb}}} \| h_\phi(\mathbf{y}_{t-w:t}) - u_t \|_2^2 + \beta_h \|\phi\|_2^2. \tag{13}$$

where $\mathbf{y}$ denotes the projection of $y$ onto its xy-components (as the control task is planar, since the robot operates with only 2 degrees of freedom in the task space, making 2D input sufficient for the inverse control mapping).

### D. Baseline: end-to-end on real data only

For comparison, we also train a baseline network $h_{\phi^{\text{base}}}$ (same architecture and loss as $h_\phi$) *using only* $\mathscr{D}_{\text{real}}$, with objective:

$$\mathscr{L}_{\text{base}}(\phi^{\text{base}}) = \sum_{(u_t, y_t)\in\mathscr{D}_{\text{real}}} \| h_{\phi^{\text{base}}}(\mathbf{y}_{t-w:t}) - u_t \|_2^2 + \beta_b \|\phi^{\text{base}}\|_2^2. \tag{14}$$

## V. RESULTS AND DISCUSSION

### A. Forward Hysteresis Modeling

*a) Overview:* We train the forward model $g_\theta$ to predict auxiliary tendon displacements from a short history of the *primary* cable inputs and the distal tip pose. At each timestep, the input is a window of primary cable displacements and the tip position. The target output is the corresponding auxiliary displacement.

*b) SOFA simulation:* The continuum robot is first designed in CAD, exported as a STEP file, and then converted to a Gmsh file containing 1064 entities, 946 nodes, and 5482 elements. Four *primary* cables at $0°, 90°, 180°, 270°$ (radial offset 4.5 mm) are modeled as `CableEquality` [9]; each spans all 11 nodes (44 primary cable nodes). Auxiliary actuation is segmented into 18 sections of length 4 mm; each section has two cables (radial offset 6.0 mm) with alternating in-plane orientations, giving 36 auxiliary cables in total. The auxiliary cables use `CableActuator` [9]. Each auxiliary cable spans 2 nodes, resulting in 72 auxiliary cable nodes overall. We impose conservative physical limits during CoReAct:

$$0 \le s_t \le 5 \text{ mm} \qquad \text{(stroke bounds)}, \tag{15}$$

$$|s_t - s_{t-1}| \le 0.005 \text{ mm} \qquad \text{(smoothness per step)}, \tag{16}$$

$$|\lambda_s| \le 10 \qquad \text{(tension bounds)}. \tag{17}$$

These bounds suppress unphysical jumps and ensure numerical stability (small $\Delta s$ steps), and they constrain tensions to keep solutions physically plausible (preventing the optimizer from injecting unreal forces that would override primary actuation).

*c) Datasets collection:* We define $\mathscr{D}_{\text{real}}$ (real dataset) as the collection of data from the physical robot, consisting of the four primary cable displacements (mapped from motor encoder ticks) and the EM-measured distal tip pose $(p_x, p_y, p_z)$. For $\mathscr{D}_{\text{real}}$, we recorded 18 trajectories at 100 Hz, covering roughly $d_x \in [-22.56, 21.58]$ mm and $d_y \in [-24.99, 29.07]$ mm in tip coordinates. Sliding windows (stride 1, zero-padded) yielded $\approx$ 258k training samples. We designed three families of input trajectories, each comprising six variants, modulated by a smooth attack-sustain-decay (ASD) envelope to facilitate gradual ramp-up and ramp-down. These include: (i) planar sinusoids at various orientations, manifesting as damped straight-line oscillations; (ii) circular loops with constant angular velocity and envelope-modulated radius, resulting in inward spirals; and (iii) square loops tracing shrinking perimeters with linear segments and corners. For validation, we additionally employed elliptical loops that contract similarly during the decay phase, enhancing shape diversity. Motor encoder ticks in $[-4096, 4096]$ correspond to $\pm 180°$. We apply an affine per-channel calibration to convert raw ticks to primary cable displacements in millimeters, accounting for pulley radius and pretension. $\mathscr{D}_{\text{res}}$ (residual dataset) is generated by applying the physics-regularized QP from the methods section (Equation (8)) over all steps in $\mathscr{D}_{\text{real}}$, yielding paired $(u_t, y_t^{\text{real}}, s_t)$ tuples as detailed in Section IV-C.0.a. This dataset is used to train the forward model.

*d) Model training process:* We train a two-layer LSTM (input size 7, hidden size 128, dropout 0.1) and a linear output head with *Tanh* activation function to predict the 36 dimensional auxiliary command at the end of each window. Adam is used for optimization (learning rate $10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) with weight decay $10^{-4}$. Input is a window of 7-D feature vector (concatenation of 4 primary cable inputs and the 3-D distal tip pose, window size 100), which is normalized to [-1, 1] per channel before fed to the neural network. The network is trained with batch size 256 for 30 epochs.

*e) Closed-loop inference:* For deployment, we configure both primary and auxiliary tendons as `CableEquality` [9] in the simulator. At each simulation step, the neural network $g_\theta$ consumes the current four

**TABLE I:** Forward hysteresis modeling accuracy on unseen trajectories. (Lower values are better.)

| | RMSE (mm) | | NRMSE (%) | |
|---|---|---|---|---|
| | $x$ | $y$ | $x$ | $y$ |
| Ellipse 1 | 0.522 | 0.930 | 2.350 | 2.070 |
| Ellipse 2 | 0.431 | 1.930 | 1.190 | 3.190 |
| OOD circle | 2.030 | 2.930 | 3.880 | 4.210 |

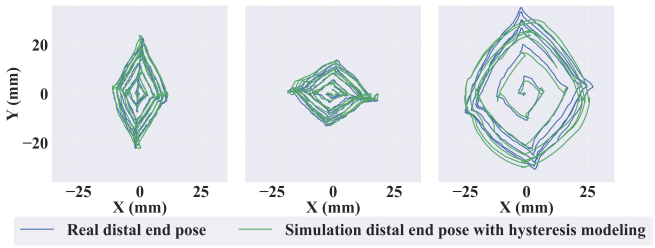


**Fig. 3: Forward model validation: predicted vs. measured tip trajectories. (a)** Ellipse 1 and **(b)** Ellipse 2 (both in-range), and **(c)** an out-of-distribution circle. Close alignment on the ellipses and preserved shape on the OOD circle indicate that the forward model captures the dominant nonlinear, path-dependent effects.

primary cable displacements and the simulated tip pose, and it outputs 36 auxiliary cable displacements, which are applied immediately as $\delta_s$, closing the loop to continually correct the simulated pose, as shown in Equation (11).

*f) Evaluation protocol and metrics:* We assess generalization on three *unseen* test trajectories: two in-distribution ellipses and one out-of-distribution (OOD) circle whose workspace exceeds the training range ($d_x \in [-25.16, 27.16]$ mm, $d_y \in [-30.74, 35.28]$ mm). For each test trajectory, we run the residual corrected digital twin in closed loop. We time-align the resulting digital twin's distal tip positions with the EM-measured ground truth at 100 Hz; if a sample is missing at a timestamp in one modality, we interpolate from neighboring points. We report per-axis RMSE (mm) and NRMSE (%):

$$\mathrm{RMSE}_a = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\big(\hat{p}_a(t) - p_a(t)\big)^2}, \qquad a \in \{x, y\}, \quad (18)$$

$$\mathrm{NRMSE}_a = \frac{\mathrm{RMSE}_a}{\max_t p_a(t) - \min_t p_a(t)} \times 100\%, \qquad a \in \{x, y\}, \quad (19)$$

where $p_a(t)$ is the ground-truth position and $\hat{p}_a(t)$ is the simulator's tip position.

*g) Results:* Figure 3 overlays the predicted and measured tip trajectories. The model tracks the in-range ellipses tightly and preserves the characteristic shape on the OOD circle. Table I shows that, on in-range ellipses, the proposed forward hysteresis modeling method achieves sub-millimeter to $\sim$2 mm error (coordinate-wise NRMSE $\leq 3.19\%$). On the much larger OOD circle, errors increase but remain moderate (RMSE up to 2.93 mm, NRMSE $\leq 4.21\%$), demonstrating extrapolation capability conferred by the physical prior.

### B. Inverse Hysteresis Compensation Control (EDIC)

*a) Overview:* We aim to learn an inverse controller $h_\phi$ that maps an ideal target trajectory to primary tendon strokes that compensate for friction, backlash, and hysteresis. Note that during training we use *imperfect* trajectories, including strong distortion caused by hysteresis, whereas at deployment EDIC receives an *ideal* target path $Y^\star = \{\mathbf{y}_t^\star\}_{t=1}^T$ (a sequence of desired distal tip poses). Formally, we seek

$$u_t \;=\; h_\phi\big(\mathbf{y}^\star_{t-w':t}\big), \quad t = 1, \ldots, T, \quad (20)$$

such that executing $u_t$ on the physical robot tracks $Y^\star$ despite hysteresis.

*b) Synthetic dataset generation:* Using CoReAct (Sec. IV-B), we generate an augmented dataset $\mathscr{D}_{\text{syn}}$ via closed-loop simulation rollouts with the learned auxiliary model: we apply $\delta_s = \hat{s}_t$ while driving novel primary sequences $u^\star_{1:T}$ through the twin. We produced 25 OOD trajectories (8 circles, 9 ellipses, 8 squares; varied amplitudes and frequencies). Simulation runs at around 70 Hz on an Intel Core i9-14900KF CPU, yielding the full synthetic set in less than 2 hours. Training the inverse model $h_\phi : \mathbb{R}^2 \to \mathbb{R}^4$ on $\mathscr{D}_{\text{real}} \cup \mathscr{D}_{\text{syn}}$ yields $\sim$645k labeled steps (258k real + 387k synthetic); the baseline uses only the 258k real steps. Both models share the same architecture and training protocol.

*c) Model and optimization:* We instantiate $h_\phi$ as a sequence-to-step LSTM regressor, conditioning on a history of planar tip positions to predict the next 4D primary tendon command. Let $\mathbf{y}_t^\star := (p_x^\star(t), p_y^\star(t)) \in \mathbb{R}^2$ be the target tip position at time $t$. Using a window of $w = 200$ steps, the input is $Y_t^\star = [\mathbf{y}^\star_{t-199}, \ldots, \mathbf{y}^\star_t] \in \mathbb{R}^{200\times 2}$ and the target output is the next primary tendon stroke $u_t \in \mathbb{R}^4$. The network is a 2-layer LSTM (hidden size 512, dropout 0.1) followed by a linear head to $\mathbb{R}^4$. We train with Adam (learning rate $10^{-3}$, weight decay $10^{-4}$) and a StepLR schedule for 30 epochs (batch size 256). Training takes around 18 min on an NVIDIA RTX 5070 Ti GPU.

**Geometry-aware metrics.** Let $\Gamma^\star$ be the ideal target curve and $\Gamma$ the realized curve. We project each point $p^\star \in \Gamma^\star$ onto $\Gamma$:

$$\hat{q}(p) := \arg\min_{q\in\Gamma} \| p^\star - q \|_2 . \quad (21)$$

Using these correspondences, we compute the average RMSE in the following way:

$$\mathrm{RMSE} = \frac{1}{2}\sum_{d\in\{x,y\}} \sqrt{\frac{1}{|\Gamma^\star|}\sum_{p\in\Gamma^\star}\big(p_d^\star - \hat{q}_d(p)\big)^2}. \quad (22)$$

NRMSE is computed the same manner as shown in Eq. (19). To quantify how closely two trajectories overlap, we also compute the planar overlap using the polygon IoU, defined as the area of intersection over the area of union of the polygons formed by $\Gamma^\star$ and $\Gamma$. Percentage improvements are defined as $(\text{Baseline} - \text{EDIC})/\text{Baseline} \times 100$ for NRMSE and $(\text{EDIC} - \text{Baseline})/\text{Baseline} \times 100$ for IoU, so that larger values indicate greater performance gains.

*d) Experiment results:* We evaluated EDIC on trajectories including circles (diameters 40–80 mm), diamonds (diagonals 50–90 mm), and Lissajous curves (vertical amplitudes 56–72 mm). Target paths are depicted in blue, with real trajectories color-coded by Euclidean error to the nearest

**TABLE II: Ideal circle tracking.**

| Diameter (mm) | 40 | 50 | 60 | 70 | 80 |
|---|---|---|---|---|---|
| EDIC RMSE (mm) ↓ | 1.344 | 1.002 | 1.362 | 2.889 | 3.155 |
| Baseline RMSE (mm) | 1.656 | 1.078 | 2.305 | 4.598 | 5.886 |
| EDIC NRMSE (%) ↓ | 3.4 | 2.0 | 2.3 | 4.1 | 3.9 |
| Baseline NRMSE (%) | 4.1 | 2.2 | 3.8 | 6.6 | 7.4 |
| NRMSE Improvement (%) ↑ | **18.8** | **7.0** | **40.9** | **37.2** | **46.4** |
| EDIC IoU ↑ | 0.849 | 0.908 | 0.893 | 0.803 | 0.820 |
| Baseline IoU | 0.811 | 0.898 | 0.815 | 0.666 | 0.634 |
| IoU Improvement (%) ↑ | **4.7** | **1.1** | **9.6** | **20.7** | **29.2** |

**TABLE III: Ideal diamond.**

| Diagonal Length (mm) | 50 | 60 | 70 | 80 | 90 |
|---|---|---|---|---|---|
| EDIC RMSE (mm) ↓ | 1.671 | 1.347 | 1.537 | 2.411 | 3.687 |
| Baseline RMSE (mm) | 2.150 | 1.990 | 2.766 | 4.397 | 6.976 |
| EDIC NRMSE (%) ↓ | 3.3 | 2.2 | 2.2 | 3.0 | 4.1 |
| Baseline NRMSE (%) | 4.3 | 3.3 | 4.0 | 5.5 | 7.8 |
| NRMSE Improvement (%) ↑ | **22.3** | **32.3** | **44.4** | **45.2** | **47.1** |
| EDIC IoU ↑ | 0.788 | 0.860 | 0.883 | 0.860 | 0.809 |
| Baseline IoU | 0.757 | 0.814 | 0.805 | 0.747 | 0.604 |
| IoU Improvement (%) ↑ | **4.0** | **5.6** | **9.6** | **15.2** | **33.9** |

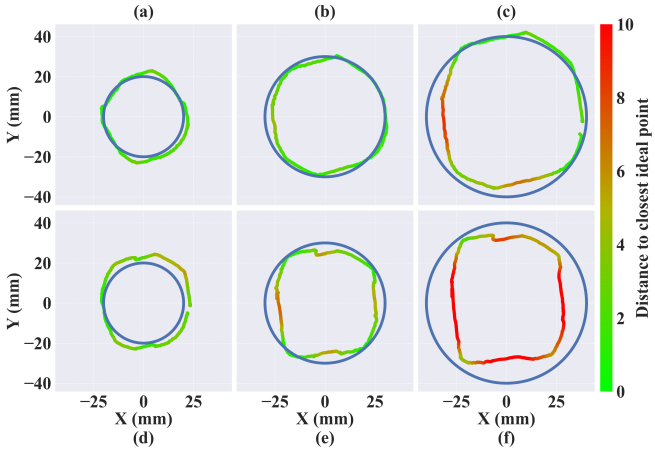


**Fig. 4: Circular targets at three scales.** Columns: 40 mm, 60 mm, 80 mm diameter; rows: EDIC (top) vs. baseline (bottom). Across scales, EDIC reduces tracking error and geometric distortion relative to the baseline.

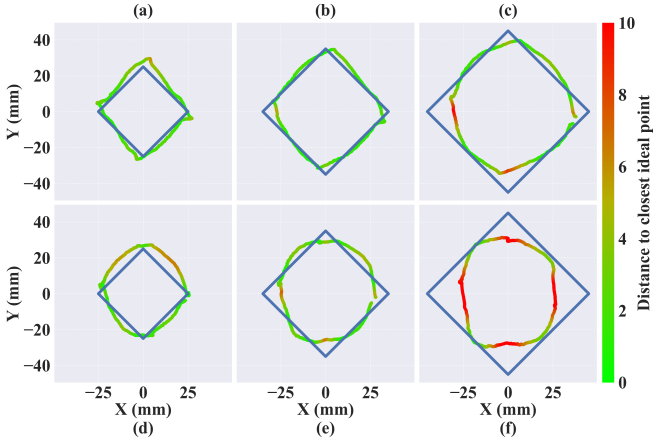


**Fig. 5: Diamond targets at increasing size.** Columns: diagonals of 50 mm, 70 mm, and 90 mm; rows: EDIC (top) vs. baseline (bottom). EDIC consistently lowers error and significantly outperforms baseline as size increases.

target point. Multi-panel figures show EDIC (top) versus baseline (bottom), with columns varying scale (Fig. 4, 5, 6).

For circular trajectories, Table II reports RMSE, NRMSE, and IoU. EDIC outperforms the baseline, with gains amplifying for larger out-of-distribution (OOD) diameters: average NRMSE reduction of 13% and IoU improvement of 3% for in-distribution (40–50 mm), escalating to around 40% and 20% for OOD (60–80 mm), peaking at 46.4% NRMSE and 29.2% IoU gains at 80 mm.

For diamond trajectories, Table III indicates consistent superiority of EDIC, with enhancements scaling with size: average NRMSE reduction of 27% and IoU gain of 5% for in-distribution (50–60 mm), rising to around 45% and 20% for OOD (70–90 mm), with maximum gains of 47.1% NRMSE and 33.9% IoU at 90 mm.

For Lissajous patterns, Table IV shows EDIC improvements: 6.6% NRMSE reduction at the boundary (56 mm), with a slight IoU dip (-3.3%); for OOD (60–72 mm), average NRMSE reduction of 32% and IoU gain of 17%, maximizing at 32.0% NRMSE and 29.1% IoU for 72 mm. EDIC demonstrates low variability across repeats (RMSE CV: 1–6%; NRMSE CV: 1–5%; IoU CV: 0.2–3%), outperforming the baseline (RMSE CV: 4–11%; NRMSE CV: 1–21%; IoU CV: 0.05–4%). Synthetic data inclusion enhances stability and repeatability.

### C. Discussion

The observed gains stem from *where* the data distribution is expanded rather than *how much*: CoReAct constrains synthesis to the reachable manifold via equality enforcement and physical bounds, yielding mechanically admissible actuation–effector pairs that curb covariate shift. EDIC's advantages increase with workspace departure and curvature (circles, diamonds, Lissajous), indicating that hybrid data aligned with feasible actuation compensates for anisotropy and friction biases that distort the baseline. Normalized nearest-point errors remain within a narrow band, implying absolute error scales with path extent while shape fidelity is preserved; correspondingly, EDIC reduces axis-aligned bias and corner rounding in large-radius and high-curvature segments, consistent with hysteresis mitigation. Training the forward twin in closed loop while deploying the inverse controller open loop amortizes physics into supervision, enabling the learned inverse to inherit actuator–workspace couplings without runtime cost. A small real corpus complemented by constrained OOD rollouts further suggests diminishing returns to unconstrained data collection relative to structure-aware synthesis targeted at coverage gaps. The proposed approach exhibits a degree of transferability. Adapting it to cable-driven continuum robots with different physical properties mainly requires rebuilding the robot-specific FEM model, configuring the corresponding parameters, choosing the optimal form of residual constraint, and collecting new data.

**TABLE IV: Ideal Lissajous.**

| Vertical Amplitude (mm) | 56 | 60 | 64 | 68 | 72 |
|---|---|---|---|---|---|
| EDIC RMSE (mm) ↓ | 1.563 | 1.823 | 2.028 | 2.480 | 3.630 |
| Baseline RMSE (mm) | 1.704 | 2.468 | 3.197 | 3.693 | 5.330 |
| EDIC NRMSE (%) ↓ | 3.0 | 3.2 | 3.3 | 3.8 | 5.3 |
| Baseline NRMSE (%) | 3.2 | 4.3 | 5.2 | 5.7 | 7.8 |
| NRMSE Improvement (%) ↑ | **6.6** | **25.4** | **36.2** | **32.8** | **32.0** |
| EDIC IoU ↑ | 0.768 | 0.759 | 0.761 | 0.748 | 0.698 |
| Baseline IoU | 0.793 | 0.721 | 0.666 | 0.634 | 0.541 |
| IoU Improvement (%) ↑ | -3.3 | **5.3** | **14.3** | **18.1** | **29.1** |

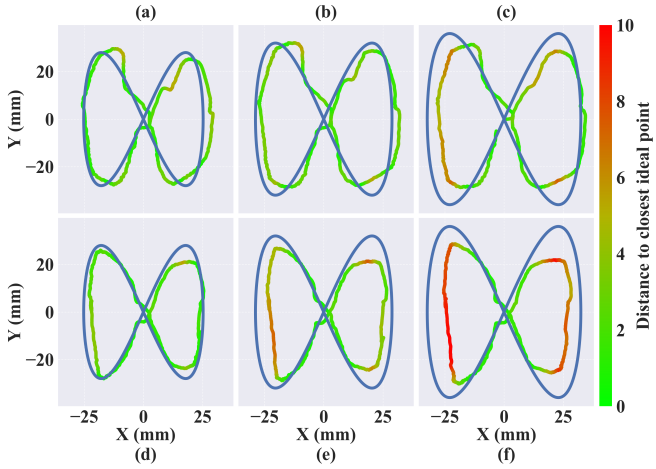


**Fig. 6: Lissajous targets with increasing amplitude.** Columns: vertical amplitude of 56 mm, 64 mm and 72 mm; rows: EDIC (top) vs. baseline (bottom). EDIC maintains accurate tracking as pattern complexity grows and outperforms the baseline at higher amplitudes.

## VI. Conclusion and Future Work

We presented a Real2Sim2Real control framework for continuum robots that integrates simulation with learning-based hysteresis compensation. By calibrating a digital twin to real data with residual actuation force, we could capture hysteretic that are not represented in standard simulations. The learned inverse kinematics model, trained on real data and data generated from this more realistic hysteresis-aware simulator, reliably maps desired tip poses to actuator commands, effectively compensating for hysteresis in real time.

Experiments demonstrate significantly improved tip accuracy and consistency, even on trajectories and configurations outside the training set. These results highlight the promise of Real2Sim2Real techniques in soft robot control: pairing high-fidelity simulations with data-driven correction yields controllers that are both physically grounded and adaptable to model uncertainties. Importantly, our results show that precise control can be achieved without a tip-mounted sensor, which is difficult to integrate in clinical settings. In future work, we plan to investigate real-time adaptive calibration, where the digital twin continually updates its residual model during operation.

## VII. Acknowledgement

This work was supported by the VP-GRACE project (Project No. 3410348), the I4.0 CONTINUUM project, the AIRCARE project funded by the Horizon Europe Programme under Grant Agreement No. 101137426, RACI project (Project No.3410430) and the Startup Grant of the Danish Institute of Advanced Study (DIAS).

## References

[1] P. Dupont, N. Simaan, H. Choset, and C. Rucker, "Continuum robots for medical interventions," *Proceedings of the IEEE*, vol. 110, no. 7, pp. 847–870, Jul 2022.

[2] D. Hanley, F. Alambeigi, and M. Khadem, "On the benefits of hysteresis in tendon driven continuum robots," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, 2025.

[3] Y. Wang *et al.*, "Using neural networks to model hysteretic kinematics in tendon-actuated continuum robots," in *Proc. Int. Symp. Med. Robot. (ISMR)*, Atlanta, USA, 2024.

[4] R. E. Goldman, A. Bajo, and N. Simaan, "Compliant motion control for multisegment continuum robots with actuation force sensing," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 890–902, 2014.

[5] D. Wu *et al.*, "Hysteresis modeling of robotic catheters based on long short-term memory network for improved environment reconstruction," *IEEE Robot. Autom. Lett.*, vol. 6, pp. 2106–2113, 2021.

[6] Y. Wang and P. E. Dupont, "Comparison of classical, neural network and hybrid models for hysteretic single-tendon catheter kinematics," *IEEE Robot. Autom. Lett. (RA-L)*, vol. 10, no. 2, pp. 1252–1259, 2025.

[7] M. Esfandiari *et al.*, "A data-driven model with hysteresis compensation for i2ris robot," June 2024, preprint.

[8] P. Ferrentino *et al.*, "Finite element analysis-based soft robotic modeling: Simulating a soft actuator in sofa," *IEEE Robotics & Automation Magazine*, vol. 31, no. 3, pp. 52–62, 2024.

[9] E. Coevoet *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.

[10] F. Faure *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft tissue biomechanical modeling for computer assisted surgery*. Springer, 2012, pp. 283–321.

[11] Y. Cho *et al.*, "Sim-to-real transfer of image-based autonomous guidewire navigation trained by deep deterministic policy gradient with behavior cloning for fast learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Kyoto, Japan, 2022, pp. 3468–3475.

[12] Y. Guo *et al.*, "Motion hysteresis compensation based on motor current segmentation for elongated cable-driven surgical instruments," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 1, pp. 721–732, Jan 2024.

[13] D. Wu *et al.*, "A review on machine learning in flexible surgical and interventional robots: Where we are and where we are going," *Biomedical Signal Processing and Control*, vol. 93, p. 106179, 2024.

[14] P. Schegg *et al.*, "Sofa gym: An open platform for reinforcement learning based on soft robot simulations," *Soft Robotics*, vol. 10, no. 2, pp. 252–261, 2023.

[15] C. Duriez, "Control of Elastic Soft Robots based on Real-Time Finite Element Method," in *2013 IEEE IEEE Int. Conf. Robot. Autom.*, 2013.

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

[17] D. Wu *et al.*, "Deep-learning-based compliant motion control of a pneumatically-driven robotic catheter," *IEEE Robot. Autom. Lett. (RA-L)*, vol. 7, no. 4, pp. 8853–8860, 2022.

[18] G. Ji *et al.*, "Towards safe control of continuum manipulator using shielded multiagent reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8434–8441, 2021.

[19] M. S. Nazeer, C. Laschi, and E. Falotico, "Rl-based adaptive controller for high precision reaching in a soft robot arm," *IEEE Transactions on Robotics*, vol. 40, no. 5, pp. 2498–2512, 2024.

[20] T. Yao *et al.*, "Sim2real learning with domain randomization for autonomous guidewire navigation in robotic-assisted endovascular procedures," *IEEE Trans. on Autom. Sci. and Eng.*, 2025.

[21] J. Gao, M. Y. Michelis, A. Spielberg, and R. K. Katzschmann, "Sim-to-real of soft robots with learned residual physics," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, p. 8523–8530, Oct. 2024.

[22] C. Duriez *et al.*, "Framework for online simulation of soft robots with optimization-based inverse model," in *Proceedings of SIMPAR 2016 conference*, Dec. 2016.