A High-Precision and Robust Geometric Relationships-Inspired Neural Network for the Inverse Kinematic Modeling of the Tendon-Actuated Continuum Manipulator

*Jinyu Duan, Jianxiong Hao, Pengyu Du, Bo Zhang, Zhiqiang Zhang, and Chaoyang Shi**

Continuum manipulators can operate in complex environments where traditional rigid manipulators fail. However, the modeling of inverse kinematics remains challenging because of its inherent nonlinearities and various external conditions. This work proposes an online learning control framework with a data cache pool utilizing a constant-curvature model inspired neural network (CCMINN) model to obtain the inverse kinematics model of tendon-actuated continuum manipulators. The CCMINN model is a kind of geometric relationships-inspired neural network, which is inspired by the geometric relationships within the constant-curvature model. This model improves the ability of traditional fully connected neural network models on high convergence speed and precision through its constant-curvature inspiration layers. These layers embed geometry insights into the neural network structure rather than loss functions like physics-informed neural networks. The online learning framework enables CCMINN to maintain high control accuracy in a variety of external load scenarios. Experiments show average tracking errors of 1.4 mm, 1.38 mm, and 1.48 mm (0.7%, 0.64%, and 0.74% of the continuum manipulator length) in the free space, under constant and variable loading conditions, respectively. The results show that combining the fast-converging CCMINN with an online learning control framework enables high-precision and robust positioning control of continuum manipulators under various external payloads.

1. Introduction

Continuum manipulators perform tasks that traditional rigid manipulators cannot achieve due to their flexible and controllable structures, smooth deformation curves, and infinite passive degrees of freedom (DOF).[1–3] These features endow continuum manipulators with excellent structural compliance, environmental adaptability, and flexible accessibility in narrow, unstructured pathways.[4–6] Therefore, continuum manipulators have been applied in complex tasks, such as aeroengine maintenance, search and rescue missions, and minimally invasive surgery.[7–14] Among various configurations of continuum manipulators, tendon-driven continuum manipulators are the most widely utilized in tasks such as thermal barrier coating repair in aeroengine combustion chambers[15] and natural orifice transluminal endoscopic surgery.[16–18] Although the trend of applying continuum manipulators across various fields is increasingly evident, challenges remain in achieving precise control in more complex and various application scenarios.[19] Inverse kinematic modeling is one of the crucial approaches for addressing these challenges. However, complex tendon routing, friction, large bending deformations, and significant hysteresis behaviors lead to the inherent nonlinearities and uncertainties, which pose substantial difficulties in inverse kinematic modeling of continuum manipulators. To overcome these difficulties and develop higher-precision and more adaptive models, model-based and data-driven approaches have been widely employed for the inverse kinematic modeling of continuum manipulators.[1,3,20,21]

J. Duan, J. Hao, P. Du, C. Shi
Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education
School of Mechanical Engineering
Tianjin University
Tianjin 300072, China
E-mail: chaoyang.shi@tju.edu.cn

B. Zhang
Future Robotics Organization
Waseda University
1 Chome-104 Totsukamachi, Shinjuku City 169-8050, Tokyo, Japan

Z. Zhang
School of Electronic and Electrical Engineering
University of Leeds
Leeds LS2 9JT, UK

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202401027.

DOI: 10.1002/aisy.202401027

The model-based inverse kinematic modeling approaches _primarily include classical geometry-based methods and mechanics-based methods.[1,3] Geometry-based methods often rely on constant-curvature (CC),[22,23] piecewise constant curvature[24,25] assumptions, or Euler curves.[26] Despite their widespread utilization, these models suffer from large errors due to the inability to satisfy CC assumptions under the complex internal friction phenomena and variable external payloads of continuum manipulators. Mechanics-based models, including Cosserat rod theory[27–29] and rigid-link models,[30–32] face difficulties in addressing the complex relationships arising from the insufficient stiffness of continuum structures. The inability of model-based methods to capture the intricate internal nonlinearities of continuum manipulators, combined with the difficulty of accurately identifying model parameters, exacerbates the errors in practical applications.[33,34] Furthermore, model-based methods are often tailored to specific configurations of continuum manipulators, limiting their generalizability and wide applicability.[10,19]

The data-driven modeling approaches have been widely used in recent research. This is because the nonlinear relationships in continuum manipulator motions are difficult to establish accurately using specific mathematical models, while data-driven modeling methods can well describe these relationships. Neural networks have been widely applied to inverse kinematic modeling of continuum manipulators because of their efficient and accurate nonlinear fitting capabilities.[35–38] In some early studies, traditional fully connected neural networks (FNNs) were employed for inverse kinematics modeling of continuum manipulators.[39–42] While FNNs improved the accuracy of inverse kinematic modeling, they struggled to handle pronounced nonlinear phenomena such as friction and hysteresis. To further enhance the accuracy of neural networks in inverse kinematic modeling, some recent studies have explored more suitable neural network architectures for continuum manipulators, such as those with time-series processing capabilities.[43–45] These architectures are well-suited for addressing nonlinearities arising from time-sequential problems such as hysteresis.

Additionally, physics-informed neural networks (PINNs)[46] are a type of neural network model that uses data to fit the desired physical model.[47,48] Their main feature is embedding physical constraints (such as partial differential equations, boundary conditions, and initial conditions) directly into the loss function of the neural network, thereby reflecting the alignment with the physical model. Some existing studies have applied PINNs for inverse kinematics modeling of continuum manipulators and have achieved good control performance.[49,50] Although PINNs further integrate the neural network with continuum manipulators' structure at the loss function level, the model still adopts conventional neural network architectures. During the training process, the model ultimately converges to the vicinity of the physical models embedded in the loss function. This characteristic means that the control methods based on PINNs are somewhat limited by the physical models' accuracy, which may lead to higher control errors compared with those methods based on traditional neural networks.[50] Moreover, all these neural network-based modeling approaches require the prior collection of datasets for training. As a result, neural networks can only fit the inverse kinematic mapping relationships of continuum manipulators based on the conditions present during data collection. When external factors such as environmental contact or changes in distal tip loads occur, the precollected datasets fail to reflect the current inverse kinematic relationships, leading to large errors in the performance of trained neural network models.

To address the above challenges of poor inverse kinematics modeling accuracy and adaptability in various external loading scenarios, a constant-curvature model inspired neural network (CCMINN) was designed. In general, CCMINN is a kind of geometric relationships-inspired neural network which is inspired by the geometric relationships within the CC model. This neural network achieves high-accuracy and high-convergence-speed inverse kinematics modeling for tendon-driven continuum manipulators. Furthermore, CCMINN was integrated into an online learning control framework with a data cache pool, enabling high-accuracy and robust distal tip position control under diverse loading conditions. The proposed CCMINN comprises an input layer, feature extraction layers, CC inspiration layers, and an output layer. The network takes the continuum manipulator's distal tip position as input and outputs the desired cable lengths. Feature extraction layers consist of two fully connected layers, each with 20 neurons, enabling CCMINN to capture statistical features from the input data. This enhances the neural network's ability to fit nonlinear relationships, thereby improving the accuracy of inverse kinematics modeling. A CC inspiration layers are designed based on the CC model. In these layers, the input and connection relationships of neurons explicitly represent the characteristic terms from the CC model, embedding geometry insights into the neural network. By incorporating this CC-based structure, the layer reduces the time required for the model to reconstruct the CC model, thereby improving the convergence speed during training. The enhanced convergence speed allows CCMINN to be further applied to an online learning control framework with a data cache pool. This pool records the continuum manipulator's historical operational data, mitigating random errors caused by frequent model updates. The combination of the fast-converging CCMINN and this robust online learning control framework enables the continuum manipulator to adapt to various loading scenarios beyond those considered during model pretraining, ensuring robust and versatile operation.

2. Experimental Section

2.1. Hardware Configuration of the Continuum Robot System

The overview of the continuum robot system has been illustrated in **Figure 1**. A spacing disk-type continuum manipulator was applied for experimental validation of the proposed approach. This continuum manipulator was operated via two pairs of symmetrically placed driving cables, which enabled bending motions in both pitch and yaw directions. The cable displacements were obtained by converting the rotary motion of the DC motors (Maxon, DC16, Switzerland) into linear motion with four linear modules. These displacements were recorded by incremental optical encoders. A 3D optical measurement unit (NDI Polaris, Ontario, Canada) measured the distal tip position of the continuum manipulator with the distally attached marker.

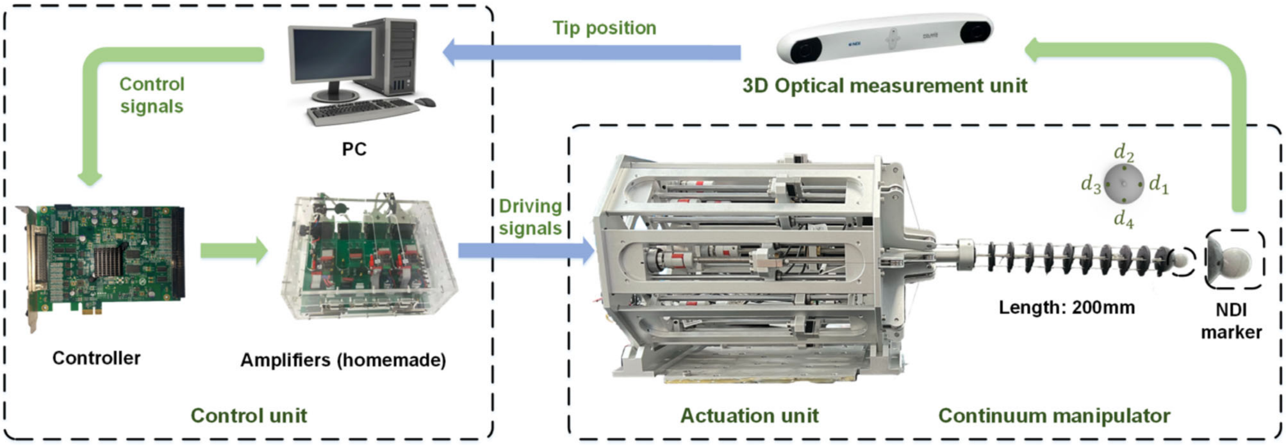


Figure 1. Hardware configuration of the continuum robot system.

The measured position data was sent to the host computer (Core i7 processor @ 2.80 GHz, NVIDIA GeForce GTX 1060 GPU @ 1.50 GHz with 6 GB VRAM, and 16 GB RAM) for positioning monitoring with a sampling rate of 60 Hz through a USB cable. The host computer calculates, analyzes, and sends control commands. A central motion controller received the control signals from the host computer and forwards these commands to the amplifiers (IMC, PENP, Germany) to drive these DC motors. These amplifiers were integrated with a homemade circuit board for more convenient system debugging and real-time control.

2.2. CC Model-Inspired Neural Network-Based Inverse Kinematic Modeling

2.2.1. CC Assumption-Based Inverse Kinematic Modeling for CCMINN Model Building

To develop the CCMINN model, the CC assumption-based inverse kinematics modeling of the continuum manipulator was initially undertaken. In our prior research,[44] the inverse kinematics model of a single-segment continuum manipulator was represented as follows

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} -\theta D\cos\varphi \\ -\theta D\sin\varphi \\ \theta D\cos\varphi \\ \theta D\sin\varphi \end{bmatrix} + s \tag{1}$$

where s, θ, and φ respectively denote the length, bending angle, and rotation angle of the continuum manipulator, d_i represents the cable displacement of the i-th driving cable, and D is the distance from the driving cable to the center of the continuum manipulator's disk.

To explicitly construct the CC-inspired structure in the proposed CCMINN model with the input data of the neural network, the distal tip position $[x, y]^T$ of the continuum manipulator needs to be extracted from (1).

According to the geometric model of a single-segment continuum manipulator, the rotation angle φ and bending angle θ satisfy the following relationship.

$$\begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{x}{r} & \frac{y}{r} & \frac{s}{R} \end{bmatrix}^T \tag{2}$$

where $r = \sqrt{x^2 + y^2}$, and R denotes the curvature radius of the continuum manipulator, as shown in (**Figure** 2b).

Substituting (2) into (1), the expression involving $[x, y]^T$ can be derived as

$$\boldsymbol{d} = \frac{sD}{rR}\boldsymbol{X} + s \tag{3}$$

Where $\boldsymbol{d} = [d_1, d_2, d_3, d_4]^T$ and $\boldsymbol{X} = [-x, -y, x, y]^T$.

Furthermore, utilizing the geometric relationships of similar triangles as depicted in (Figure 2b), the following relationship can be obtained.

$$rR = \frac{1}{2}L^2 \tag{4}$$

where L is the chord length of the continuum manipulator, and $L^2 = x^2 + y^2 + z^2$. It should be noted that variables x, y, and z are constrainted by the following geometric relationship

$$s = \frac{r^2 + z^2}{r}\operatorname{atan}\left(\frac{r}{z}\right) \tag{5}$$

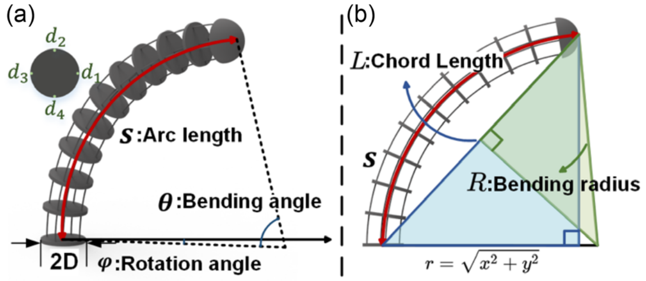


Figure 2. Illustration of a) the configuration variables and b) two similar triangles of the 2-DOF continuum manipulator bending in the 3D space.

Solving for the value of z from (5) is challenging and slow. Therefore, an efficient approximation of atan(·)[51] is applied for transforming (5) into the following cubic equation.

$$z^3 - sz^2 + r^2 z - 0.281sr^2 = 0 \tag{6}$$

The above equation allows to be efficiently calculated z directly utilizing the root-finding formula of a cubic equation, eliminating the need to solve (5) iteratively, which will also accelerate the computation of the neural network model in the following section. Meanwhile, the slight error introduced by this approximation can be indirectly compensated during subsequent neural network training.

Substituting (4) into (3), the inverse kinematics expression of the single-segment continuum manipulator, which is expressed explicitly in terms of $\boldsymbol{X}$, can be derived as

$$\boldsymbol{d} = \left(\frac{2sD}{L^2} + k\right)\boldsymbol{X} - (k\boldsymbol{X} - s) \tag{7}$$

where k is a hyperparameter to be learned and does not require a specific predefinition. This formula intuitively illustrates the algebraic relationship between the inputs $\boldsymbol{X}$ and outputs $\boldsymbol{d}$ of the CCMINN to be designed, making it relatively straightforward to integrate the CC model into the neural network. This integration improved the neural network model's accuracy while enhancing its convergence speed. Introducing k not only explicitly reflects the linear relationship between $\boldsymbol{d}$ and $\boldsymbol{X}$ but also increases the learnable parameters in the neural network model, thereby enhancing the model's learning capability.

2.2.2. CCMINN-Based Neural Network Architecture for Inverse Kinematic Modeling

Based on the derived CC model above, the CCMINN model was proposed to realize high-accuracy inverse kinematic modeling with high convergence speed. According to (7), the CCMINN model can be designed and structured as illustrated in **Figure 3**.

The proposed CCMINN model comprises four main components: the input layer, feature extraction layers, CC inspiration layers, and the output layer. The input layer takes the desired distal tip position ($\boldsymbol{X} \in \mathbb{R}^{4\times1}$) as input and forwards this information to the feature extraction layers. These layers include two fully connected layers containing 20 neural cells, and the internal parameters and hyperparameters should be trained or tuned to optimize feature extraction from the output of the input layer. The extracted features were then passed on to the CC inspiration layers that primarily consist of the nonlinear-inspired part (NLIP) and the linear-inspired part (LIP). These two parts embody the inverse kinematic mapping relationship as depicted in (7) through neural cell structures, specifically

$$\begin{aligned} \boldsymbol{O_N} &= f_N\left(\frac{2sD}{L^2}\boldsymbol{W}_1 \cdot \boldsymbol{J}\right) + f_N(\boldsymbol{W}_2 \cdot \boldsymbol{J}), \\ \boldsymbol{O_L} &= \boldsymbol{W}_2 \cdot \boldsymbol{J} - s, \boldsymbol{\gamma} = \boldsymbol{O_N} - \boldsymbol{O_L} \end{aligned} \tag{8}$$

where $\boldsymbol{O_N}$ and $\boldsymbol{O_L}$ respectively denote the output of NLIP and LIP, f_N represents the activation function of NLIP and LIP, respectively. In this work, tanh was chosen as f_N. $\boldsymbol{J}$expressed the output of the feature extraction layers. $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ were weights of these layers; these weights were treated as hyperparameter k in (7), and the form of parameters $2sD/L^2$ and s was inspired by the geometry relationship represented in (7). After obtaining $\boldsymbol{\gamma}$ by subtracting $\boldsymbol{O_N}$ from $\boldsymbol{O_L}$, the CC inspiration layers passed the result in the form of (7) through two fully connected

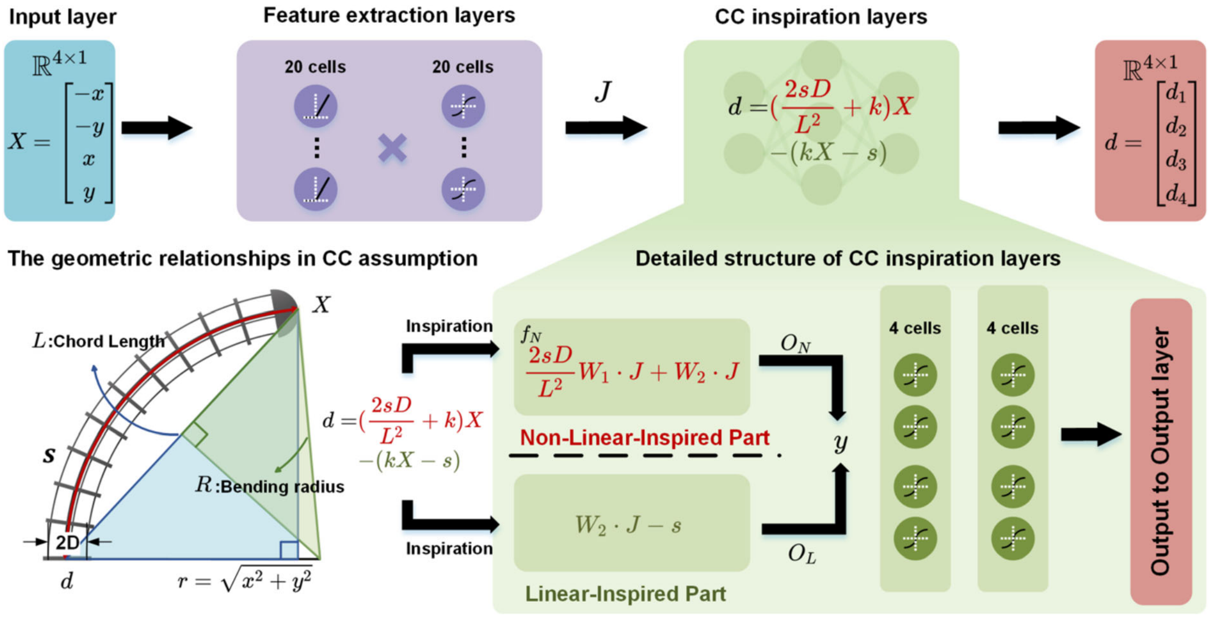

Input layer
Feature extraction layers
CC inspiration layers
$\mathbb{R}^{4\times1}$
$X = \begin{bmatrix} -x \\ -y \\ x \\ y \end{bmatrix}$
20 cells
20 cells
J
$d = (\frac{2sD}{L^2} + k)X - (kX - s)$
$\mathbb{R}^{4\times1}$
$d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}$
The geometric relationships in CC assumption
Detailed structure of CC inspiration layers
L:Chord Length
X
s
R:Bending radius
2D
d
$r = \sqrt{x^2 + y^2}$
Inspiration
$d = (\frac{2sD}{L^2} + k)X$
$-(kX - s)$
Inspiration
f_N
$\frac{2sD}{L^2} W_1 \cdot J + W_2 \cdot J$
Non-Linear-Inspired Part
O_N
γ
$W_2 \cdot J - s$
O_L
Linear-Inspired Part
4 cells
4 cells
Output to Output layer

Figure 3. Inverse kinematic modeling based on the proposed CCMINN model.

layers, each with four neurons, to generate an output. The output layer received the input from the CC inspiration layers and ultimately predicted cable displacements ($\boldsymbol{d} \in \mathbb{R}^{4\times1}$). Hence, the CCMINN model establishes a mapping relationship from the distal tip position to the cable displacements of the continuum manipulator.

Further analysis of (8) reveals that the proposed CC inspiration layers not only explicitly incorporate prior knowledge of the system but also enhance the convergence speed of the CCMINN model through their excellent gradient properties. The gradients of $\boldsymbol{\gamma}$ with respect to the weights $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are

$$\begin{aligned}\frac{\partial\boldsymbol{\gamma}}{\partial\boldsymbol{W}_1} &= \left(1-\tanh^2\left(\frac{2sD}{L^2}\boldsymbol{W}_1\cdot\boldsymbol{J}\right)\right)\cdot\frac{2sD}{L^2}\cdot\boldsymbol{J},\\ \frac{\partial\boldsymbol{\gamma}}{\partial\boldsymbol{W}_2} &= (1-\tanh^2(\boldsymbol{W}_2\cdot\boldsymbol{J}))\cdot\boldsymbol{J}-\boldsymbol{J}\end{aligned} \tag{9}$$

For the weight $\boldsymbol{W}_1$, $2sD/L^2$ is a relatively small value in the tendon-driven continuum manipulators model ($L \gg D$), which keeps tanh(·) in the nonsaturated region. This enables $\boldsymbol{W}_1$ to update stably with a relatively large gradient. Meanwhile, for the weight $\boldsymbol{W}_2$, the presence of $-\boldsymbol{J}$ provided a stable gradient update path. Even when the gradient contribution of the term $(1-\tanh^2(\cdot))$ diminished, $\boldsymbol{W}_2$ can still update stably. These excellent gradient properties further enhanced the convergence speed of the CCMINN model.

The proposed CCMINN model explicitly incorporates prior geometric knowledge through its CC inspiration layers. Compared with traditional neural networks (e.g., traditional FNN model), this approach ensured high accuracy in inverse kinematic modeling while significantly reducing training convergence time. This accelerated convergence enabled the CCMINN model to be effectively applied within the control framework with online learning of continuum manipulators, thereby enhancing its robustness and control precision, as will be discussed in the following section.

To verify that the proposed CCMINN model has advantages in convergence speed compared with traditional neural networks (such as feedforward neural networks, FNNs) while ensuring high accuracy in the training process, offline training experiments on the CCMINN model and the FNN model were conducted. The dataset utilized came from an experiment with a 200 mm continuum manipulator. The continuum manipulator's distal tip was driven to move along six circle trajectories with different diameters (from Φ36 mm to Φ216 mm, with an interval of 36 mm). The distal tip position data were collected with the 3D optical measurement unit, and the tendon displacement data were recorded simultaneously. Finally, a total of 4,025 sets of data were obtained, which were randomly divided into training sets (80%) and test sets (20%). The loss function utilized in the training process is the mean absolute percentage error, and its calculation formula is shown in (9).

$$\text{Loss} = \frac{100}{n}\sum\frac{|y_{\text{pre}}-y_{\text{true}}|}{y_{\text{true}}} \tag{10}$$

where y_{pre} and y_{ture}, respectively, denote the prediction from the neural network and ground truth.

The loss value variation and details of each neural network's training processes are presented in **Figure 4** and **Table 1**. The loss function value of the proposed CCMINN model finally reached 0.12, which was significantly lower than 0.22 of FNN. Comparing the convergence speed of the two network models, the number of training epochs required for CCMINN to reach 1% loss and 0.5% loss was 6 and 43, respectively, which was only 46.15% and 17.55% of FNN. This advantage is due to the geometry inspiration layer based on the CC model introduced by CCMINN, which not only has a faster convergence speed but also has a lower initial loss function value (1.28, which was 1.29% of FNN). It should be noted that the loss curve of FNN in the first 10 epochs had a higher rate of change (9.22/epoch) because its higher initial loss function value brings a higher gradient value, which had nothing to do with the convergence speed of FNN itself. In those epochs, which have the same range of loss function values, such as 1–40 epochs of CCMINN and 13–249 epochs of FNN (loss value from 0.99 to 0.22), the loss curve of CCMINN had a higher convergence rate (0.013/epoch, which was 4.36 times faster than that of the FNN). The experimental result indicates that the proposed CCMINN model has advantages in convergence speed compared with traditional neural networks

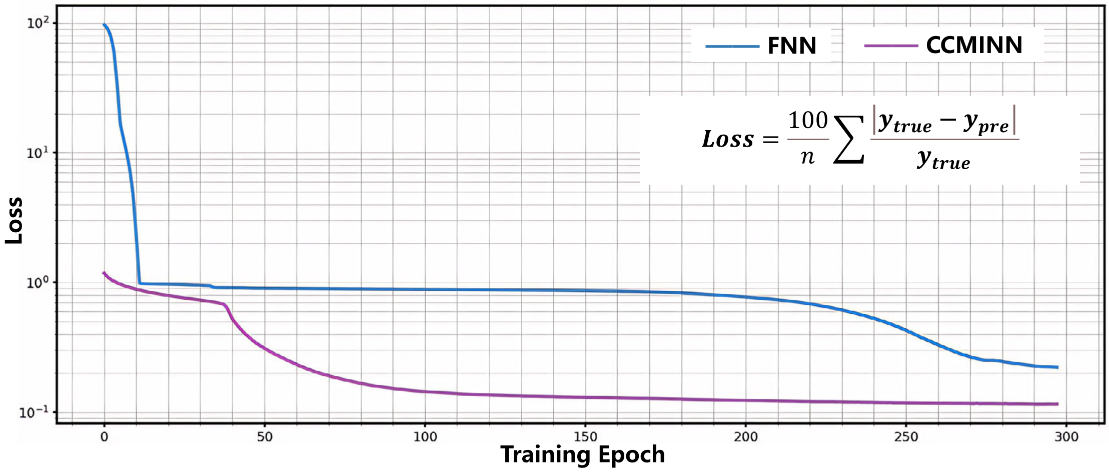


Figure 4. The variations of the loss function while training CCMINN and FNN in the offline training experiments.

Table 1. Offline training experimental result of CCMINN and FNN.

Training metric	CCMINN	FNN
Initial loss value	1.28	99.52
Final loss value	0.12	0.22
Epochs to reach 1% loss	6	13
Epochs to reach 0.5% loss	43	245
Convergence rate (loss value from 0.99 to 0.22)	0.013/epoch	0.003/epoch

Notes: 1% loss and 0.5% loss represent the loss value which equal to 1 and 0.5, respectively.

2.3. CCMINN-Based Online Learning Control Framework for the Continuum Manipulator

A control framework that integrates an online learning method for continuum manipulators was designed based on the proposed CCMINN model to improve the robustness and precision of this model across different operating environments. The overview of this control framework is shown in **Figure 5**. This control framework consists of three main modules: the CCMINN module, the online learning quality monitoring module, and the data cache module. The CCMINN module takes the desired distal tip position $\boldsymbol{X_d}$ as input and utilizes the CCMINN model to output predicted cable displacements $\boldsymbol{d}$, which control the continuum manipulator's distal tip to generate the actual distal tip position $\boldsymbol{X_a}$. The online learning quality monitoring module determines the framework's operational mode by calculating the Euclidean distance between the desired and actual distal tip positions. If this distance is below a predefined threshold, the framework operates in online learning mode; otherwise, it switches to offline application mode. The different modes primarily affect the operation of the data cache module. During the online learning mode, the data cache module recorded real-time tendon displacements $\boldsymbol{d}$ (output of the CCMINN module) and measured distal tip positions $\boldsymbol{X_m}$ of the continuum manipulator as input–output pairs (IO Pairs) in a data cache pool. This cache pool stored the continuum manipulator's historical operational data, helping to mitigate random errors caused by frequent model updates. When the capacity controller detects that the data cache pool has reached a specified threshold, it shuffles and utilizes the recorded data to conduct online learning on the CCMINN model using the follow the regularized leader algorithm.[52] It should be highlighted that all the recorded data reflected the current inverse kinematic mapping relationship. Thus, this update made the model parameters better fit the current inverse kinematic relationship of the continuum manipulator, enhancing the robustness and precision of the CCMINN model in various execution environments. In the offline application mode, the data cache module does not interact with the CCMINN module, as the CCMINN model can control the actual continuum manipulator's distal tip movement within a specified error threshold.

By utilizing the CCMINN-based control framework, when changes occurred in the actual inverse kinematic relationship of the continuum manipulator (e.g., due to variations in the distal tip loads), the fast-converging CCMINN model effectively updated to adapt to the latest mapping relationship. This capability enabled the control framework to achieve high robustness and precision in positioning tasks for the continuum manipulator across various execution environments.

3. Experiments and Results

3.1. Trajectory Tracking of the Continuum Manipulator Based on the Proposed CCMINN-Based Control Framework in the Free Space

In this section, two groups of trajectory tracking experiments were conducted on a continuum manipulator in free space to validate the feasibility of integrating the proposed CCMINN model into an online learning control framework. Initially, the high accuracy of this control framework is confirmed through

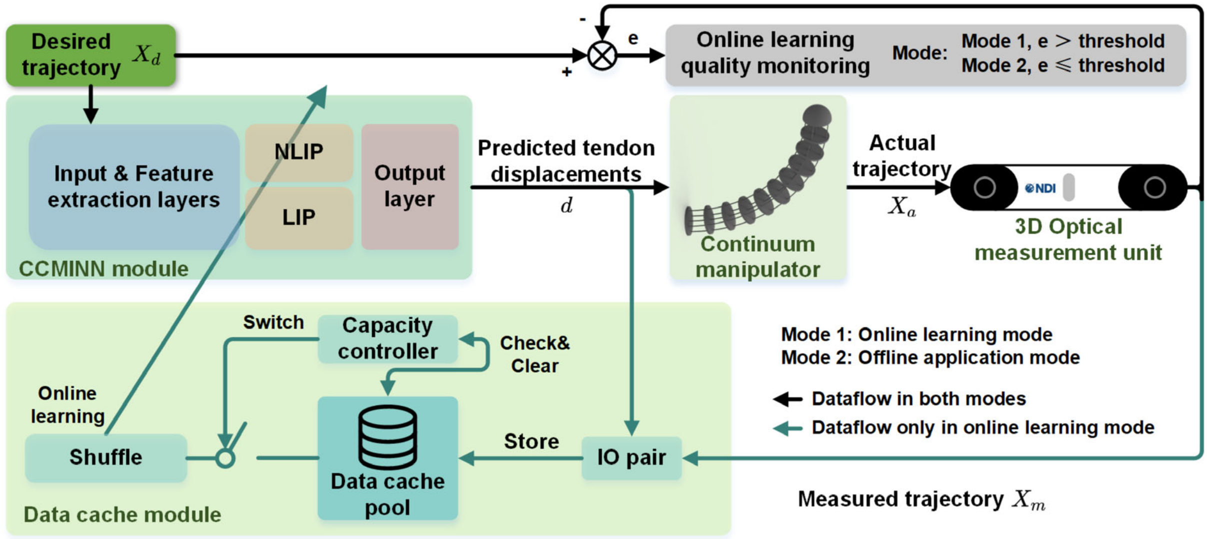


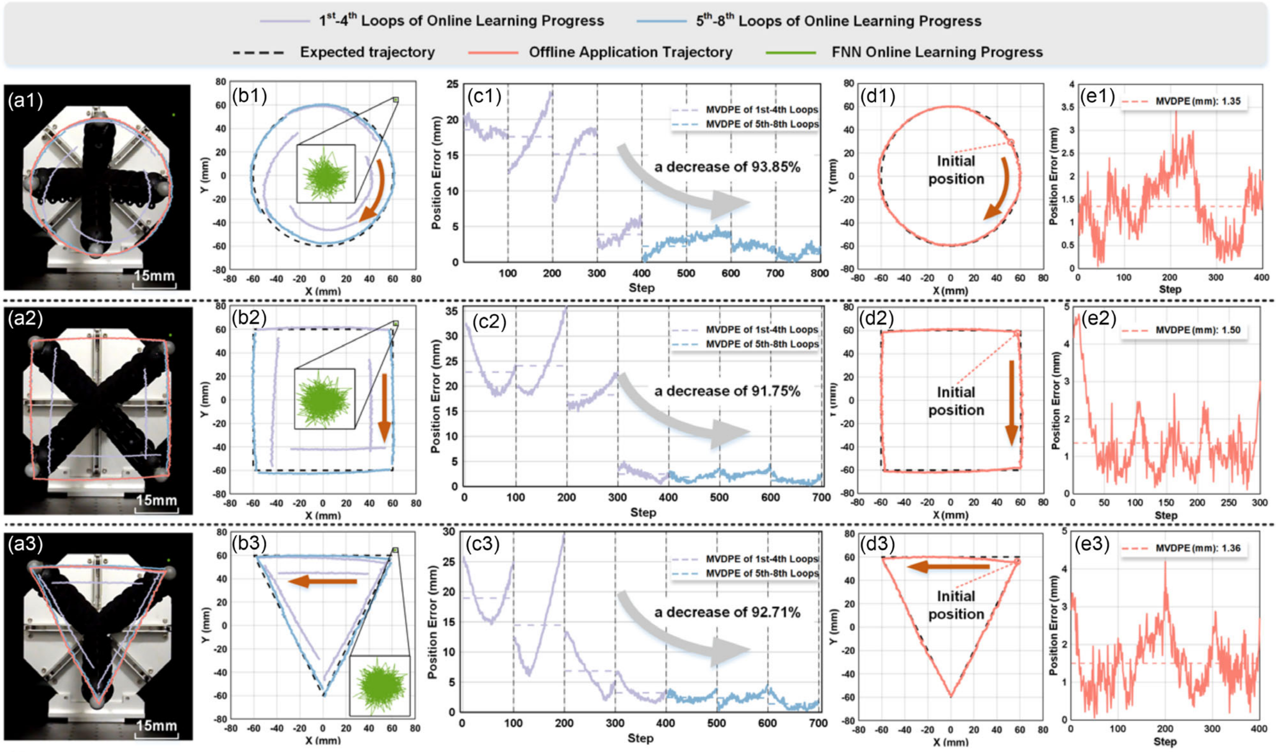
Figure 5. The proposed online learning framework with a data cache pool which is applied to the distal tip position control of continuum manipulators.

the trajectory tracking experiment using the newly online learning fast-converging CCMINN model. Furthermore, the transferability of this model is also assessed by the experiments with trajectories that differ from those used in training. The mean value of the distal positioning error (MVDPE), which compares the actual trajectory produced by the CCMINN-based control framework to the desired trajectory, serves as the key performance metric in these experiments.

3.1.1. Trajectory Tracking Experiment with the de novo Online Learning CCMINN Model

To validate the feasibility of the proposed rapid-converging CCMINN model within the online learning control framework as outlined in the previous section, experiments were carried out on three distinct desired trajectories. To avoid potential hardware damage to the continuum robot from completely random model outputs, this experiment used 100 sets of data generated by the CC assumption-based model to pretrained the model with only one epoch. For each trajectory, the data cache pool was set with 100 sets of data, and the online learning was halted when the relative MVDPE reached a threshold of 1%. Once online learning was stopped, an open-loop trajectory tracking task was performed for the same desired trajectory. Each trajectory was repeated three times. Throughout the experiments, the MVDPE trend in the CCMINN model's online learning phase and the MVDPE results for the open-loop trajectory tracking task were recorded. The corresponding experimental results are presented in **Figure 6** and **Table 2**.

In contrast, the traditional FNN model, when applied within the same online learning framework, failed to complete the online learning task. This model produced a large average MVDPE value of 95.96 mm, which is 47.98% of the length of the continuum manipulator, during the online learning process. It should be pointed out that the variation in MVDPE for the FNN model in the experiment mainly came from the change of the desired trajectory because the distal tip of the continuum manipulator remained in almost the same position. This failure is attributed to the slower convergence speed of the traditional FNN model. The large amount of singular data from the unconverged model caused the continuum manipulator to remain in the same distal tip position under the constraints of limit protection (as illustrated in Figure 6b1–b3). As a result, the data cache pool within the online learning framework accumulated a large amount of repetitive data for model training, further hindering the training process of the traditional FNN model.



Notes:
1. The step volume of each loop of the online learning progress is set as 100.
2. Online learning is triggered once every 1/4 of the trajectory (every 1/3 of triangle trajectory).

Figure 6. The tracking performances and the MVDPE variations of the proposed CCMINN model and FNN model in the trajectory tracking experiments in the free space. a1–a3) The physical pictures on circle trajectory, rectangle trajectory, and triangle trajectory; b1–b3) the tracking performances and details of each model during the online learning progress on circle trajectory, rectangle trajectory, and triangle trajectory; c1–c3) the MVDPE variations of each model during the online learning progress on circle trajectory, rectangle trajectory, and triangle trajectory; d1–d3) the tracking performances and details of each model in the offline application on circle trajectory, rectangle trajectory, and triangle trajectory; and e1–e3) the MVDPE of each model after the online learning progress on circle trajectory, rectangle trajectory, and triangle trajectory.

Table 2. Trajectory tracking experimental results of different models based on the proposed online control framework in the free space.

Desired trajectory	Method	FRMVDPE	LRMVDPE	RMVDPE Reduction	MVDPE [mm]	RMVDPE	MaxVDPE [mm]	RMaxVDPE
Circle	CCMINN	9.27%	0.57%	93.85%	1.35	0.67%	3.42	1.71%
	FNN	50.07%	50.11%	/	/	/	/	/
Rectangle	CCMINN	11.40%	0.94%	91.75%	1.50	0.75%	4.20	2.10%
	FNN	51.49%	51.53%	/	/	/	/	/
Triangle	CCMINN	9.47%	0.69%	92.71%	1.36	0.68%	4.80	2.40%
	FNN	42.33%	42.36%	/	/	/	/	/
Mean value on three trajectories	CCMINN	10.05%	0.73%	92.74%	1.40	0.70%	4.14	2.07%
	FNN	47.96%	48.00%	/	/	/	/	/

Note: RMVDPE and RMaxVDPE represent relative mean value of distal positioning error and relative max value of distal positioning error, respectively. FRMVDPE and LRMVDPE, respectively, denote the mean value of distal positioning error of the first and last loop in the online learning progress for the proposed CCMINN model (the first and third loop in the online learning process for the FNN model).

However, during the online learning process, the MVDPE value for the continuum manipulator decreased across all desired trajectories, with an average decrease of 92.74% across three trajectories, eventually reaching the threshold for halting online learning, which was a relative MVDPE of 1%. This demonstrates that the proposed CCMINN model can be efficiently implemented within the presented online learning framework because of its high convergence speed. Furthermore, in the open-loop trajectory tracking experiments that followed the online learning phase, the MVDPE of the continuum manipulator reached 1.40 mm (0.70% of the manipulator's length). This large improvement indicates that the CCMINN model, once trained within this framework, can be applied with high precision in the kinematic control of the continuum manipulator. This situation highlights the advantage of the CCMINN model's rapid convergence in online learning applications.

3.1.2. Transferability of the CCMINN-Based Control Framework on the Other Trajectories

To assess the transferability of the CCMINN model across various control trajectories different from those used in training, three groups of transferability experiments were carried out. The CCMINN models trained through online learning in *Experiment 3.1.1 Trajectory Tracking Experiment with the de novo Online Learning CCMINN Model* were utilized for open-loop trajectory tracking tasks on new trajectories. Each trajectory was repeated three times. The MVDPE results of the trajectory tracking experiments were recorded, and the experimental results are illustrated in **Figure 7** and **Table 3**.

It is evident that the continuum manipulator maintained a low MVDPE when utilizing the transferred CCMINN model. On average, across these three experiments, the MVDPE was 1.49 mm, which is 0.75% of the length of the continuum manipulator. This highlights that the proposed CCMINN model and the CCMINN-based online control framework possess strong transferability and are not limited to the initial training trajectories. A comprehensive analysis of the results from *Experiments 3.1.1* and *3.1.2* for the same CCMINN model is detailed in Table 3. For models trained on square trajectories, the transferred MVDPE decreased from 1.50 mm to 1.41 mm, while for those trained on circular trajectories, the transferred MVDPE increased from 1.35 to 1.55 mm. This variation is attributed to the complexity of the desired trajectories, the square the most complex and the circle the simplest. More complex trajectories contain more mapping information, which allows the model to better learn the rich mapping relationship. This underscores the importance of selecting appropriately complex training trajectories to ensure model convergence and achieve the desired precision in the final application.

3.2. Trajectory Tracking of the Continuum Manipulator Based on the Proposed CCMINN-Based Control Framework with External Loading

In this section, two groups of experiments were carried out on the continuum manipulator platform to demonstrate the strong robustness of the proposed CCMINN-based control framework under both constant and variable external loading conditions because of its ability to update the model parameters according to current measured data.

3.2.1. Trajectory Tracking Experiment Under Constant Loads

The pretrained CCMINN model in the free space from *Experiment 3.1.1 Trajectory Tracking Experiment with the de novo Online Learning CCMINN Model* was further engaged in online learning tasks for three new desired trajectories. The data cache size was set equal to the number of trajectory points, and the threshold for terminating online learning was consistent with that used in *Experiment 3.1.1.* During this experiment, a weight of 50 g was suspended at the distal tip of the continuum manipulator, as illustrated in **Figure 8**a1. Each trajectory was repeated three times. The experimental results are analyzed and presented in Figure 8 and **Table 4**.

At the outset of the experiment, the continuum manipulator exhibited a high MVDPE value, averaging 13.91 mm across the three trajectories, which is 6.96% of the manipulator's length. This error was due to the pretrained CCMINN model being

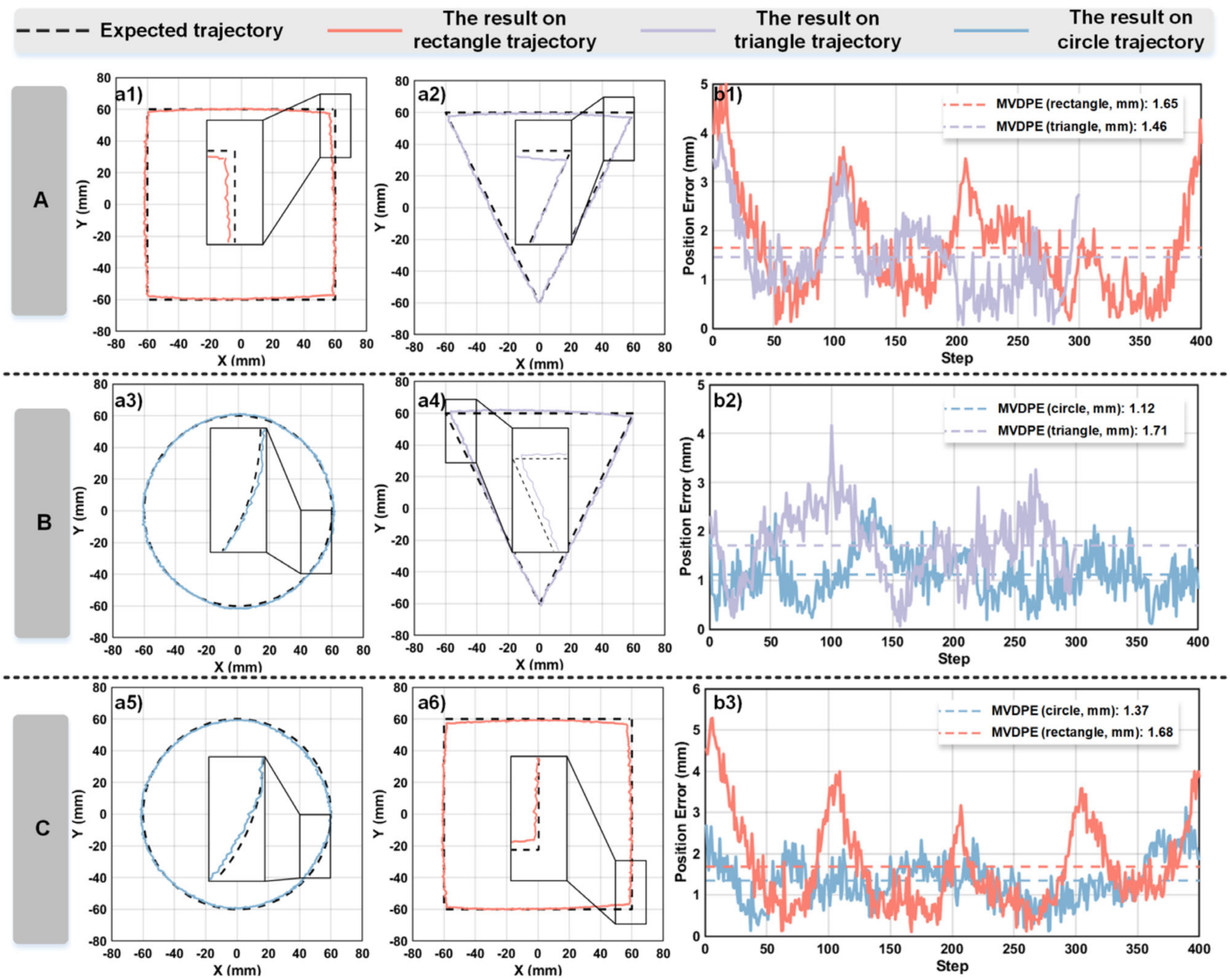
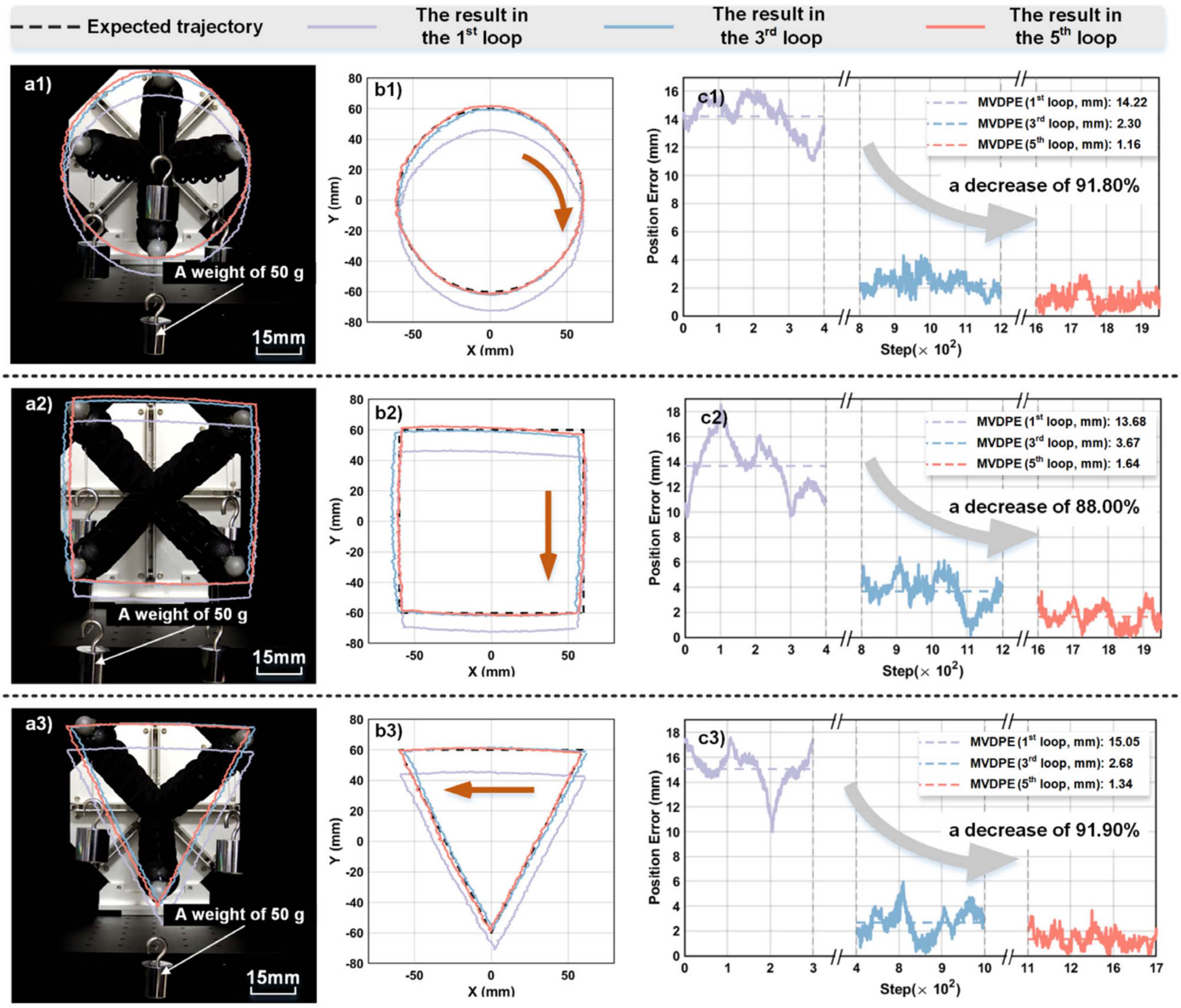


Figure 7. The tracking performances and the tracking MVDPE variations of the proposed CCMINN model in the free space. a1–a6) The tracking performances and details of each trajectory; b1–b3) the MVDPE variations of each trajectory; (a1,a2) the tracking performances and details of CCMINN model trained on circle trajectory; (a3,a4) the tracking performances and details of CCMINN model trained on rectangle trajectory; and (a5,a6) the tracking performances and details of CCMINN model trained on triangle trajectory. Label A, B and C respectively denote the experiments using the CCMINN model trained on circle trajectory, rectangle trajectory and triangle trajectory in Experiment 3.1.1.

Table 3. Trajectory tracking experimental results of transferability experiments in the free space.

Training trajectory	Desired trajectory	MVDPE [mm]	RMVDPE	MaxVDPE [mm]	RMaxVDPE
Circle	Rectangle	1.65	0.82%	4.99	2.50%
	Triangle	1.46	0.73%	3.97	1.98%
	Mean value on two trajectories	1.55	0.78%	4.48	2.24%
Rectangle	Circle	1.12	0.56%	2.66	1.33%
	Triangle	1.71	0.85%	4.17	2.08%
	Mean value on two trajectories	1.41	0.71%	3.42	1.71%
Triangle	Circle	1.35	0.67%	3.12	1.56%
	Rectangle	1.68	0.84%	5.29	2.64%
	Mean value on two trajectories	1.51	0.76%	4.21	2.10%

Notes: RMVDPE, MaxVDPE, and RMaxVDPE represent relative mean value of distal positioning error, max value of distal positioning error, and relative max value of distal positioning error, respectively.



Notes:
1. The step volume of each loop of the online learning progress is set equal to the number of trajectory points.
2. Online learning occurs once every full trajectory.

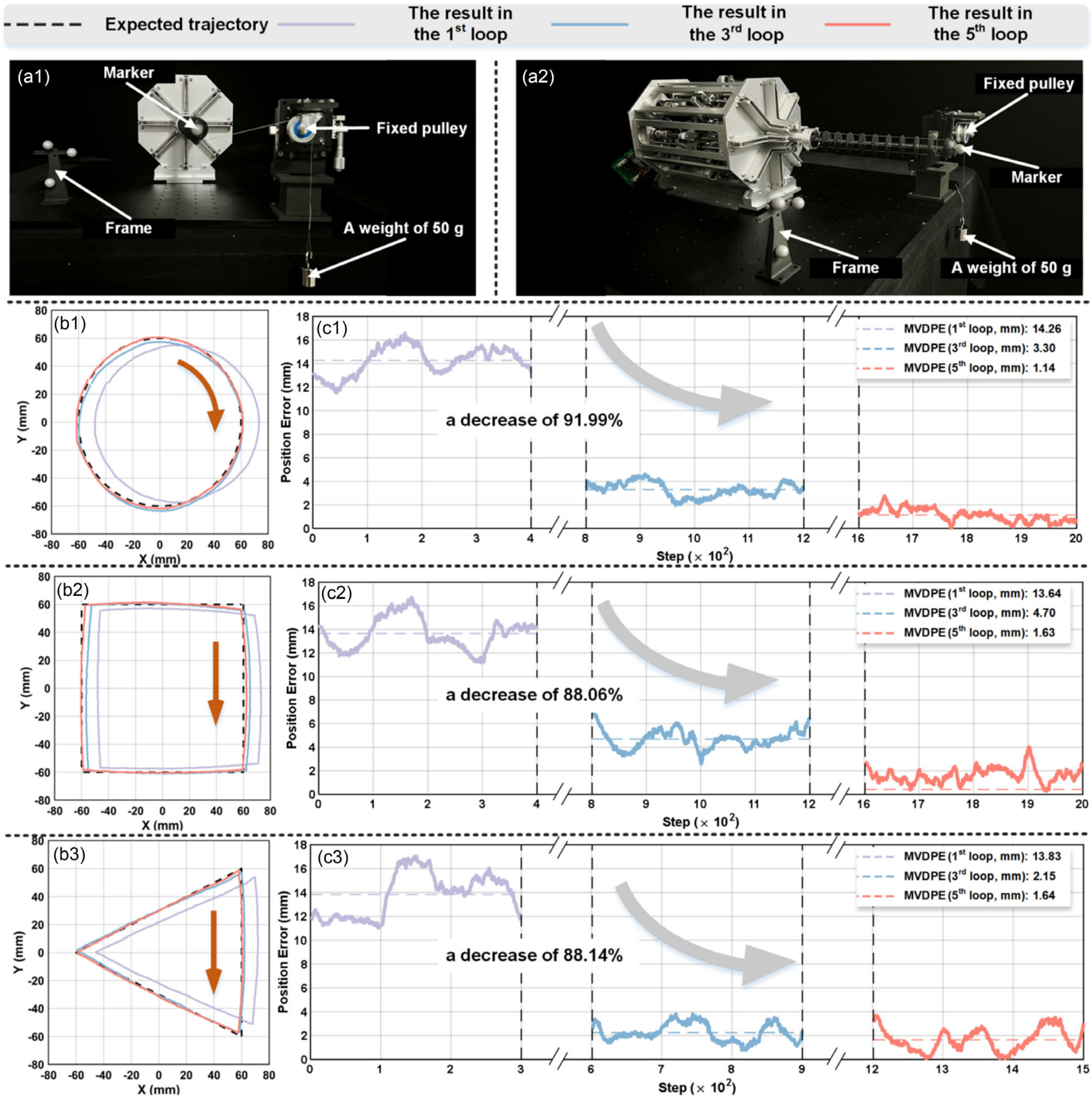
Figure 8. The tracking performances and the MVDPE variations of the proposed CCMINN model in the trajectory tracking experiments under a constant load. a1–a3) The physical pictures on circle trajectory, rectangle trajectory, and triangle trajectory; b1–b3) the tracking performances and details during the online learning progress on circle trajectory, rectangle trajectory, and triangle trajectory; and c1–c3) the MVDPE variations during the online learning progress on circle trajectory, rectangle trajectory, and triangle trajectory.

Table 4. Trajectory tracking experimental results of different models based on the proposed online control framework under constant loads.

Desired trajectory	FMVDPE [mm]	FRMVDPE	FMaxVDPE [mm]	FRMaxVDPE	LMVDPE [mm]	LRMVDPE	LMaxVDPE [mm]	LRMaxVDPE	RMVDPE Reduction
Circle	14.22	7.11%	16.15	8.07%	1.17	0.58%	2.91	1.46%	91.8%
Rectangle	13.68	6.84%	18.58	9.29%	1.64	0.82%	3.67	1.84%	88.00%
Triangle	15.05	7.52%	17.56	8.78%	1.34	0.67%	3.67	1.83%	91.9%
Mean value on three trajectories	14.32	7.16%	17.43	8.72%	1.38	0.69%	3.42	1.71%	90.34%

Notes: RMVDPE, MaxVDPE, and RMaxVDPE represent relative mean value of distal positioning error, max value of distal positioning error, and relative max value of distal positioning error, respectively; FRMVDPE and LRMVDPE, respectively, denote the mean value of distal positioning error of the first and last loop in the online learning progress for the proposed CCMINN model.

calibrated to the inverse kinematics of the continuum manipulator in free space. When the distal load was altered, the inverse kinematics relationship shifted, resulting in significant positional errors at the distal tip. As online learning progressed, the relative MVDPE for each trajectory decreased significantly, with an average reduction of 90.34% across the three trajectories, ultimately reaching the predefined threshold for ending online learning at 1%. This large reduction is attributed to the framework's capability to dynamically update the CCMINN model's parameters with current cable displacements and distal tip positions. Moreover, CCMINN has a sufficiently fast convergence speed, enabling the model to quickly respond to adjustments



Notes:
1. The step volume of each loop of the online learning progress is set equal to the number of trajectory points.
2. Online learning occurs once every full trajectory.

Figure 9. The tracking performances and the MVDPE variations of the proposed CCMINN model in the trajectory tracking experiments under a variable load. a1,a2) The physical pictures experimental setup; b1–b3) the tracking performances and details during the online learning progress on circle trajectory, rectangle trajectory, and triangle trajectory; and c1–c3) the MVDPE variations during the online learning progress on circle trajectory, rectangle trajectory, and triangle trajectory.

Table 5. Trajectory tracking experimental results of different models based on the proposed online control framework under variable loads.

Desired trajectory	FMVDPE [mm]	FRMVDPE	FMaxVDPE [mm]	FRMaxVDPE	LMVDPE [mm]	LRMVDPE	LMaxVDPE [mm]	LRMaxVDPE	RMVDPE Reduction
Circle	14.26	7.13%	16.58	8.29%	1.14	0.57%	2.77	1.39%	91.99%
Rectangle	13.64	6.82%	16.72	8.36%	1.63	0.81%	4.03	2.01%	88.06%
Triangle	13.83	6.92%	17.03	8.52%	1.64	0.82%	3.68	1.84%	88.14%
Mean value on three trajectories	13.91	6.96%	16.78	8.39%	1.48	0.74%	3.49	1.75	89.39%

Notes: RMVDPE, MaxVDPE, and RMaxVDPE represent relative mean value of distal positioning error, max value of distal positioning error, and relative max value of distal positioning error, respectively; FRMVDPE and LRMVDPE, respectively, denote the mean value of distal positioning error of the first and last loop in the online learning progress for the proposed CCMINN model.

made to its parameters by the framework. The updated CCMINN model can then accurately reflect the current inverse kinematics of the continuum manipulator, thereby reducing the MVDPE value. The experimental outcomes demonstrate that the proposed CCMINN-based control framework can effectively update the fast-converging CCMINN model's parameters to match the current inverse kinematics of the continuum manipulator under a constant distal load. This advantage makes the control framework possess high robustness, maintaining high distal positioning accuracy even when the load conditions differ from those of the model's pretraining environment.

3.2.2. Trajectory Tracking Experiment Under Variable Loads

Unlike *Experiment 3.2.1 Trajectory Tracking Experiment under Constant Loads*, a weight of 50 g was attached to the distal tip of the manipulator through a fixed pulley in this experiment, as shown in **Figure 9**a1,a2. This configuration resulted in the force direction applied to the manipulator's distal tip continuously changing throughout its motion. Each trajectory was repeated three times. The experimental results are illustrated in Figure 9 and **Table 5**. Despite the changing distal load, it was observed that the relative MVDPE value for each desired trajectory of the continuum manipulator decreased as online learning progressed, with an average reduction of 89.39% across the three trajectories, and ultimately reached the preset threshold for terminating online learning at 1%. This indicates that the proposed CCMINN-based control framework can effectively update its model parameters to accommodate the varying inverse kinematics of the continuum manipulator under changing loads. Together with the results from the previous experiment in this section, it can be concluded that the proposed CCMINN-based control framework is capable of swiftly adjusting the rapid-converging CCMINN model's parameters via online learning to align with the current inverse kinematics of the continuum manipulator under different distal load conditions. This ensures robust and precise distal positioning control.

3.3. Discussion

The experimental results indicate that integrating the proposed fast-converging CCMINN model with an online learning control framework can effectively achieve high-precision and high-robustness control tasks for the distal tip position of continuum manipulators under various motion trajectories and load conditions. Moreover, the superiority of the CCMINN model compared to the FNN model has been demonstrated in experiments. Through the explicit incorporation of prior geometric knowledge based on a CC model in the CC inspiration layers, the proposed CCMINN model possesses a rapid convergence speed. This fast convergence speed allows the CCMINN model to be applied in an online learning framework, generating high-precision and strongly robust positioning control results. The data cache pool in the online learning framework records the continuum manipulator's historical operational data, mitigating random errors caused by frequent model updates. During the experiments, the volume of this pool was set to 100. While a larger cache volume can record more operational data, thereby enhancing the model training accuracy, it also increases data collection time and online learning time. This time increase may lead to the control frameworks delayed response to changes in the inverse kinematics of the continuum manipulator, which will increase errors to some extent. Therefore, the appropriate cache volume size needs to be determined in practical applications. In transferability experiments (*Experiment 3.1.2*), it was observed that when the proposed CCMINN model is trained on simple trajectories and applied to complex ones, the error increases accordingly; conversely, the error decreases when trained on complex trajectories and applied to simpler ones. This phenomenon suggests that the complexity of training trajectories should be reasonably determined when formulating model training tasks to avoid application failures due to overly simplistic training models.

4. Conclusion

A fast-converging CCMINN model is proposed to be incorporated into an online learning control framework with a data cache pool, generating high-precision and strongly robust positioning control performance of continuum manipulators under various external payloads. The CC inspiration layers of this CCMINN model combine the geometry knowledge of the CC model and the neural network's ability to fit nonlinear relationships, which makes this model achieve high-convergence-speed and high-accuracy inverse kinematics modeling for tendon-driven continuum manipulators. Moreover, the proposed online learning control framework updates the rapid-converging CCMINN model with the recorded current inverse kinematics data, helping it fit the precise inverse kinematics mapping relationships under various external loading conditions. Additionally, the data cache pool in this control framework records the continuum manipulator's historical operational data, reducing random errors caused by frequent model updates, which benefits the high-precision control tasks. In this work, the CC inspiration layers are based on the CC assumption of single-segment continuum manipulators. Converting the geometry model into an inverse kinematics model of multisegment continuum manipulators will extend this fast-converging CCMINN model and online learning control framework to address high-precision and robust positioning control of multisegment continuum manipulators. Future research will also incorporate multimodal sensing sources, including tension force and distal tip contact force, into the CCMINN model to enhance the efficacy of inverse kinematics modeling in unstructured environments. Fiber Bragg grating-based sensing techniques[53–55] will be developed and integrated to capture multimodal sensing information, combining shape and force data.

Acknowledgements

J.D. and J.H. contributed equally to this work. This work was supported in part by the National Natural Science Foundation of China under grant nos. 92148201 and 52475029, and Royal Society under IEC\NSFC\211360. This work was also supported by the International Institute for Innovative Design and Intelligent Manufacturing of Tianjin University in Zhejiang, Shaoxing 312000, CN.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

continuum manipulators, inverse kinematic modeling, online learning, physics-informed neural networks, trajectory tracking

Received: November 28, 2024
Revised: May 16, 2025
Published online:

[1] R. J. Webster, B. A. Jones, *Int. J. Rob. Res.* **2010**, *29*, 1661.
[2] C. Laschi, B. Mazzolai, M. Cianchetti, *Sci. Robot.* **2016**, *1*, eaah3690.
[3] J. Zhang, Q. Fang, P. Xiang, D. Sun, Y. Xue, R. Jin, K. Qiu, R. Xiong, Y. Wang, H. Lu, *Cyborg Bionic Syst* **2022**, *2022*, https://spj.science.org/doi/10.34133/2022/9754697.
[4] S. Kim, C. Laschi, B. Trimmer, *Trends in Biotechnology* **2013**, *31*, 287.
[5] M. Russo, S. M. H. Sadati, X. Dong, A. Mohammad, I. D. Walker, C. Bergeles, K. Xu, D. A. Axinte, *Advanced Intelligent Systems* **2023**, *5*, 2200367.
[6] N. El-Atab, R. B. Mishra, F. Al-Modaf, L. Joharji, A. A. Alsharif, H. Alamoudi, M. Diaz, N. Qaiser, M. M. Hussain, *Adv. Intell. Syst.* **2020**, *2*, 2000128.
[7] J. Burgner-Kahrs, D. C. Rucker, H. Choset, *IEEE Trans. Robot.* **2015**, *31*, 1261.
[8] S. K. Talas, B. A. Baydere, T. Altinsoy, C. Tutcu, E. Samur, *Soft Rob.* **2020**, *7*, 521.
[9] M. C. Lastinger, S. Verma, A. D. Kapadia, I. D. Walker, in *2019 Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Montreal, QC, Canada **2019**, pp. 5365–5371.
[10] J. Kim, M. De Mathelin, K. Ikuta, D.-S. Kwon *Proc. IEEE* **2022**, Vol. *110*, pp. 909–931.
[11] M. Pan, C. Yuan, X. Liang, T. Dong, T. Liu, J. Zhang, J. Zou, H. Yang, C. Bowen, *Adv. Intell. Syst.* **2022**, *4*, 2100140.
[12] J. Li, S. Wang, Z. Zhang, C. Shi, *IEEE Robot. Automat. Magn.* **2024**, *32*. 2.
[13] J. Li, D. Wu, J. Hao, B. Zhang, C. Hu, C. Shi, *IEEE Trans. Instrum. Meas.* **2025**, *74*, 1.
[14] D. Song, X. Yu, B. Leng, B. Zhang, C. Shi, *IEEE/ASME Trans. Mechatron.* **2025**, 1. https://doi.org/10.1109/TMECH.2025.3563471.
[15] X. Dong, M. Wang, A. Mohammad, W. Ba, M. Russo, A. Norton, J. Kell, D. Axinte, *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 4217.
[16] D. Song, S. Wang, Z. Zhang, X. Yu, C. Shi, *IEEE Trans. Med. Robot. Bionics* **2023**, *5*, 657.
[17] J. Zhu, L. Lyu, Y. Xu, H. Liang, X. Zhang, H. Ding, Z. Wu, *Adv. Intell. Syst.* **2021**, *3*, 2100011.
[18] Z. Lin, Z. Xu, H. Liu, X. Wang, X. Ai, C. Zhou, B. Huang, W. Chen, G.-Z. Yang, A. Gao, *Adv. Intell. Syst.* **2024**, *6*, 2300373.
[19] T. Da Veiga, J. H. Chandler, P. Lloyd, G. Pittiglio, N. J. Wilkinson, A. K. Hoshiar, R. A. Harris, P. Valdastri, *Prog. Biomed. Eng.* **2020**, *2*, 032003.
[20] A. Bajo, N. Simaan, *Int. J. Rob. Res.* **2016**, *35*, 422.
[21] M. S. Xavier, A. J. Fleming, Y. K. Yong, *Adv. Intell. Syst.* **2021**, *3*, 2000187.
[22] Z. Li, L. Wu, H. Ren, H. Yu, *Mech. Machine Theory* **2017**, *107*, 148.
[23] Y. Li, D. H. Myszka, A. Murray, *IEEE Robot. Autom. Lett.* **2023**, *8*, 7631.
[24] X. Huang, J. Zou, G. Gu, *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 3175.
[25] P. Rao, Q. Peyron, S. Lilge, J. Burgner-Kahrs, *Front. Rob. AI* **2021**, *7*, 630245.
[26] P. Rao, Q. Peyron, J. Burgner-Kahrs, in *2021 IEEE Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Xi'an, China **2021**, pp. 1402–1408.
[27] M. Tummers, V. Lebastard, F. Boyer, J. Troccaz, B. Rosa, M. T. Chikhaoui, *IEEE Trans. Rob.* **2023**, *39*, 2360.
[28] F. Boyer, V. Lebastard, F. Candelier, F. Renda, M. Alamir, *IEEE Trans. Rob.* **2023**, *39*, 1544.
[29] J. Duan, K. Zhang, K. Qian, J. Hao, Z. Zhang, C. Shi, *Cyborg. Bionic. Syst.* **2024**, *5*, 0110.
[30] D. C. Rucker, R. J. Webster Iii, *IEEE Trans. Rob.* **2011**, *27*, 1033.
[31] R. J. Roesthuis, S. Misra, *IEEE Trans. Rob.* **2016**, *32*, 372.
[32] T. S. Lee, E. A. Alandoli, *J Braz. Soc. Mech. Sci. Eng.* **2020**, *42*, 508.
[33] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, F. Renda, *IEEE Trans. Robot.* **2023**, *39*, 1728.
[34] T. G. Thuruthel, Y. Ansari, E. Falotico, C. Laschi, *Soft Rob.* **2018**, *5*, 149.
[35] X. Zhou, Z. Xu, S. Li, H. Wu, T. Cheng, X. Lv, *AI Based Robot Safe Learning And Control*, Springer, Singapore **2020**.
[36] X. Wang, Y. Li, K.-W. Kwok, *Front. Rob. AI* **2021**, *8*, 730330.
[37] N. Liang, R. M. Grassmann, S. Lilge, J. Burgner-Kahrs, in *2021 IEEE Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Xi'an, China **2021**, pp. 1387–1393.

[38] T. G. Thuruthel, E. Falotico, M. Cianchetti, C. Laschi, *ROMANSY 21 - Robot Design, Dynamics and Control* (Eds: V. Parenti-Castelli, W. Schiehlen) Vol. *569*, Springer International Publishing, Cham **2016**, pp. 47–54.

[39] T. G. Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, C. Laschi, *Soft Rob.* **2017**, *4*, 285.

[40] A. Ghoul, K. Kara, S. Djeffal, M. Benrabah, M. L. Hadjili, *Arch. Mech. Eng.* **2022**, *69*, 595.

[41] I. M. Loutfi, A. H. B. Boutchouang, A. Melingui, O. Lakhal, F. B. Motto, R. Merzouki, *IFAC-PapersOnLine* **2020**, *53*, 9899.

[42] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, C. Laschi, *Bioinspir. Biomim.* **2015**, *10*, 035006.

[43] N. Tan, P. Yu, F. Ni, Z. Sun, in *2021 IEEE Inter. Conf. on Systems, Man, and Cybernetics (SMC)*, IEEE, Melbourne, Australia **2021**, pp. 1035–1041.

[44] J. Hao, J. Duan, K. Wang, C. Hu, C. Shi, *IEEE Trans. Med. Rob. Bionics* **2023**, *5*, 916.

[45] Y. Ansari, E. Falotico, Y. Mollard, B. Busch, M. Cianchetti, C. Laschi, in *2016 6th IEEE Inter. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, IEEE, Singapore, Singapore **2016**, pp. 457–463.

[46] M. Raissi, P. Perdikaris, G. E. Karniadakis, *Journal of Computational Physics* **2019**, *378*, 686.

[47] J. Zhang, Y. Zhao, F. Shone, Z. Li, A. F. Frangi, S. Q. Xie, Z.-Q. Zhang, *IEEE Trans. Neural Syst. Rehabil. Eng.* **2023**, *31*, 484.

[48] M. Bensch, T.-D. Job, T.-L. Habich, T. Seel, M. Schappler, in *2024 IEEE Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Yokohama, Japan **2024**, pp. 17293–17299.

[49] W. Sun, N. Akashi, Y. Kuniyoshi, K. Nakajima, *IEEE Robot. Autom. Lett.* **2022**, *7*, 6862.

[50] T. Yoon, Y. Chai, Y. Jang, H. Lee, J. Kim, J. Kwon, J. Kim, S. Choi, *IEEE Robot. Autom. Lett.* **2024**, *9*, 3068.

[51] S. Rajan, Sichun Wang, R. Inkol, A. Joyal, *IEEE Signal Process. Mag.* **2006**, *23*, 108.

[52] B. McMahan in *Proc. of the Fourteenth Inter. Conf. on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, (Eds. G. Gordon, D. Dunson, M. Dudík) Vol. *15*, PMLR, Fort Lauderdale, FL, USA **2011**, pp. 525-533.

[53] Y. Hao, H. Zhang, Z. Zhang, C. Hu, C. Shi, *IEEE Trans. Med. Robot. Bionics* **2024**, *6*, 868.

[54] Z. Tang, S. Wang, M. Li, C. Shi, *IEEE Trans. Med. Robot. Bionics* **2022**, *4*, 145.

[55] C. Shi, X. Luo, J. Guo, Z. Najdovski, T. Fukuda, H. Ren, *IEEE J. Biomed. Health Inform.* **2018**, *22*, 806.