Precision Mechanics & Digital Fabrication
https://www.acadlore.com/journals/PMDF

Forward Kinematics Solution for Cable-Driven Hyper-Redundant Manipulators Based on BiLSTM

Tianao Wang, Zhenghao Liang, Guolei Wang*

Department of Mechanical Engineering, Tsinghua University, 100084 Beijing, China

* Correspondence: Guolei Wang (wangguolei@mail.tsinghua.edu.cn)

Received: 01-26-2025 **Revised:** 03-02-2025 **Accepted:** 03-07-2025

Citation: T. A. Wang, Z. H. Liang, and G. L. Wang, "Forward kinematics solution for cable-driven hyper-redundant manipulators based on BiLSTM," *Precis. Mech. Digit. Fabr.*, vol. 2, no. 1, pp. 44–56, 2025. https://doi.org/10.56578/pmdf020104.

Abstract: This paper investigates the kinematic solution of cable-driven hyper-redundant manipulators, focusing on the transformation from the cable-driven space to the joint space. Two forward kinematics solution networks based on residual networks and bidirectional long short-term memory (BiLSTM) networks are proposed and compared. First, a single-joint kinematic model is established based on the topology of the cable-driven hyper-redundant manipulator, providing the mapping relationship between cable length variations and joint angles. The decoupling problem between the cable-driven space and joint space is analyzed, extending the decoupling method from a two-joint scenario to a multi-joint scenario, leading to the derivation of coupled equations between cable lengths and joint angles. Subsequently, both a single-joint forward kinematics solution network and a multi-joint forward kinematics solution network are designed and trained separately. Finally, their performance is evaluated using a test dataset. The results demonstrate that the multi-joint forward kinematics solution network significantly outperforms the single-joint network in terms of both accuracy and computational efficiency.

Keywords: Cable-driven; Hyper-redundant manipulator; Kinematics; Residual network; Long short-term memory (LSTM) network

1 Introduction

Industrial robots play a crucial role in various application scenarios, significantly improving production efficiency and safety [1]. However, traditional rigid manipulators have large volumes and limited degrees of freedom, making them difficult to adapt to complex or narrow environments. In such specific environments, cable-driven hyper-redundant manipulators exhibit better adaptability [2]. Their driving method is based on rear-mounted motors, with a separation between the electromechanical system, providing high flexibility and mobility [3]. Therefore, it has been widely used in tasks such as disaster rescue [4], aerospace [5], aircraft assembly [6], nuclear industry [7] and surgery [8].

Cable-driven hyper-redundant manipulators are actuated by motors pulling steel cables. Besides the mapping relationship between the workspace and joint space, there is also a mapping relationship between the cable-driven space and the joint space [9]. Compared with traditional manipulators, cable-driven hyper-redundant manipulators possess more degrees of freedom, making their forward kinematics modeling process more complex. Typically, such problems involve complex nonlinear equations, where the calculation process is not only cumbersome and slow but may also result in multiple solutions [10]. This paper focuses on exploring the mapping relationship between the cable-driven space and the joint space.

Xu et al. [11] established a cable coupling relationship and proposed a decoupling method to compensate for the coupled motion between cables. The above studies indicate that the kinematic equations between the cable-driven space and the joint space can be solved using both analytical and numerical methods.

Neural networks have shown significant advantages in handling highly nonlinear problems [12, 13]. By employing multilayer nonlinear processing units, they can effectively capture and model complex nonlinear relationships. Takatani et al. [14] achieved the kinematic solution for hyper-redundant manipulators by combining multiple neural network models. Dong et al. [15] established a pre-trained neural network model to obtain its inverse kinematic solution with

high accuracy. The above studies demonstrate that artificial neural networks perform better than traditional parametric models when solving the kinematics of cable-driven hyper-redundant manipulators.

Due to the highly coupled nature of the driving cables in cable-driven hyper-redundant manipulators, the coupling becomes increasingly complex as the number of joints increases. However, there still exist certain relationships between the cables among different joints and between different cables within the same joint. To address this issue, this paper proposes two deep learning-based network models to process the forward kinematic relationship between the cable-driven space and the joint space. Specifically, both networks adopt residual networks [16] and LSTM networks [17]. Through training and comparison, the model that converges and achieves high-precision forward kinematic solutions is selected.

The structure of this paper is arranged as follows: Section 2 establishes the forward kinematics model between the cable-driven space and the joint space based on the decoupling method. Section 3 proposes a single-joint forward kinematics solution network and a BiLSTM-based forward kinematics network for the overall manipulator. Section 4 analyzes the effectiveness of different networks through comparative experiments. Finally, conclusions are drawn.

2 Forward Kinematics Modeling of the Cable-Driven Space and Joint Space Based on the Decoupling Method

The forward kinematics model of the cable-driven space and joint space mainly aims to solve the mapping relationship between the joint angles and the cable lengths of the cable-driven hyper-redundant manipulator. Taking an 8-joint manipulator as an example, due to the coupling relationship between cables, in order to decouple and simplify the forward kinematics analysis of the cable-driven space and joint space, it is necessary to first establish a single-joint kinematics model to solve the mapping relationship between three cables l_i and two joint angles ψ and α. Then, based on the sequential connection between joints, the multi-joint forward kinematics decoupling analysis is carried out using the single-joint kinematic equation.

2.1 Analysis of the Mapping Relationship Between Single-Joint Cable Length Variation and Joint Angles

The single joint of a cable-driven hyper-redundant manipulator is mainly composed of a universal joint, forming a parallel mechanism. Its inputs are the variations of three cables l_i, and the outputs are the two-degree-of-freedom joint angles. By changing the lengths of the three cables l_i between the joint disks, two degrees of freedom can be controlled, thereby enabling movement in three-dimensional space. To describe the mapping relationship between cable variations and joint angles, this paper establishes a kinematic model for a single joint, as shown in Figure 1.

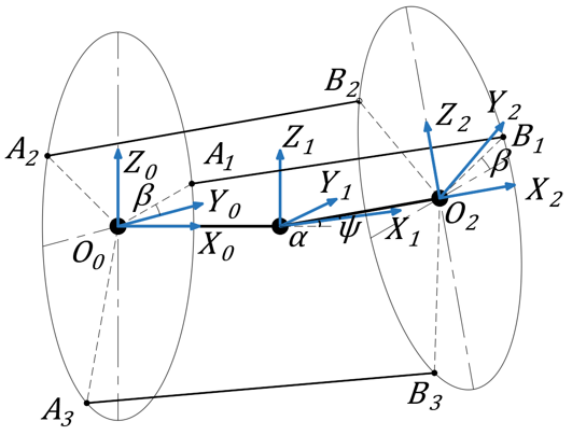

Figure 1. Single-joint kinematics model analysis diagram

The plane $A_1A_2A_3$, centered at O_0, represents joint disk 1, while the plane $B_1B_2B_3$, centered at O_2, represents joint disk 2. The lines A_1B_1, A_2B_2, and A_3B_3 represent the three cables l_1, l_2, and l_3 that drive the joint motion. Three coordinate systems $\{0\}$, $\{1\}$, and $\{2\}$ are established at the center of disk $1O_1$, the single-joint center, and the center of disk $2O_2$, respectively. The Z-axis directions of coordinate systems $\{0\}$ and $\{2\}$ are along the robot link axis, while the X -axis and Y -axis directions correspond to the two rotational degrees of freedom of the joint. Since β represents the position of the cable on the routing disk, it satisfies $\beta = \angle BO_2Y_2 = \angle AO_1Y_1$.

The transformation relationship from coordinate system $\{0\}$ to coordinate system $\{2\}$ can be obtained by solving the homogeneous transformation matrix from coordinate system $\{0\}$ to coordinate system $\{1\}$ and the homogeneous transformation matrix from coordinate system $\{1\}$ to coordinate system $\{2\}$. Let the distance between coordinate system $\{0\}$ and coordinate system $\{2\}$ be d, then the distance between the disk and the joint center point is $d/2$.

Coordinate system $\{0\}$ first translates along the X_0-axis direction by $d/2$, then rotates around the Z-axis of the translated coordinate system by angle α to obtain coordinate system $\{1\}$. Its homogeneous transformation matrix is shown in Eq. (1):

$$
{}^{0}_{1}T = \mathrm{Trans}_{X,\frac{d}{2}}\,\mathrm{Rot}_{Z,\alpha} = \begin{bmatrix} c\alpha & -s\alpha & 0 & d/2 \\ s\alpha & c\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

Coordinate system $\{1\}$ rotates around the Y_1-axis by angle ψ, then translates along the X-axis direction of the rotated coordinate system by $d/2$. The homogeneous transformation matrix is shown in Eq. (2):

$$
{}^{1}_{2}T = \mathrm{Rot}_{Y,\psi}\,\mathrm{Trans}_{X,\frac{d}{2}} = \begin{bmatrix} c\psi & 0 & s\psi & \frac{d}{2}c\alpha \\ 0 & 1 & 0 & 0 \\ -s\psi & 0 & c\psi & -\frac{d}{2}s\alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

Thus, the homogeneous transformation matrix between coordinate system $\{0\}$ and coordinate system $\{2\}$ is shown in Eq. (3):

$$
{}^{0}_{2}T = {}^{0}_{1}T\,{}^{1}_{2}T = \begin{bmatrix} c\alpha c\psi & -s\alpha & c\alpha s\psi & \frac{d}{2} + \frac{d}{2}c\alpha c\psi \\ c\psi s\alpha & c\alpha & s\alpha s\psi & \frac{d}{2}s\alpha c\psi \\ -s\psi & 0 & c\psi & \frac{d}{2}s\psi \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}
$$

For cables A_1B_1, A_2B_2, and A_3B_3, the only difference lies in the position parameter β of the cable on the routing disk. Therefore, β is used as the position parameter to obtain the general solution. Here, cable A_1B_1 is taken as an example. A_1 in coordinate system $\{0\}$ can be expressed as Eq. (4):

$$
A_1^0 = \begin{bmatrix} 0 \\ rc_{\beta 1} \\ rs_{\beta 1} \\ 1 \end{bmatrix} \tag{4}
$$

B_1 in coordinate system $\{2\}$ can be expressed as Eq. (5):

$$
B_1^2 = \begin{bmatrix} 0 \\ rc_{\beta 1} \\ rs_{\beta 1} \\ 1 \end{bmatrix} \tag{5}
$$

Using Eq. (3), B_1 in coordinate system $\{0\}$ can be expressed as Eq. (6):

$$
B_1^0 = {}^{0}_{2}T\,B_1^2 = \begin{bmatrix} \frac{d}{2} - rs\alpha c_{\beta 1} + \frac{d}{2}c\alpha c\psi - rc\alpha s\psi s_{\beta 1} \\ rc\alpha c_{\beta 1} + \frac{d}{2}s\alpha c\psi - rs\alpha s\psi s_{\beta 1} \\ \frac{d}{2}s\psi + rc\alpha s_{\beta 1} \\ 1 \end{bmatrix} \tag{6}
$$

Thus, the length of cable A_1B_1, denoted as l_1, is given by Eq. (7):

$$
l_1 = |A_1B_1| = \left| \begin{array}{c} \frac{d}{2} - rs\alpha c_{\beta 1} + \frac{d}{2}c\alpha c\psi - rc\alpha s\psi s_{\beta 1} \\ rc\alpha c_{\beta 1} + \frac{d}{2}s\alpha c\psi - rs\alpha s\psi s_{\beta 1} - rc_{\beta 1} \\ \frac{d}{2}s\psi + rc\alpha s_{\beta 1} - rs_{\beta 1} \\ 1 \end{array} \right| \tag{7}
$$

Therefore, by providing the two-degree-of-freedom joint angles of the joint, the lengths of the three cables can be solved. Conversely, given the lengths of the three cables, the two-degree-of-freedom joint angles can be determined, thereby achieving the forward kinematics analysis of the single-joint cable-driven space and joint space.

The cable length calculation formula is given in Eq. (8):

$$\begin{aligned} f(\psi,\alpha,\beta) = \Bigg(& \left(\frac{d}{2} - rs\alpha c_\beta + \frac{d}{2}c\alpha c\psi - rc\alpha s\psi s_\beta\right)^2 \\ & + \left(\frac{d}{2}s\psi + rc\alpha s_\beta - rs_\beta\right)^2 \\ & + \left(rc\alpha c_\beta + \frac{d}{2}s\alpha c\psi - rs\alpha s\psi s_\beta - rc_\beta\right)^2 \Bigg)^{1/2} \end{aligned} \tag{8}$$

2.2 Decoupling Analysis of the Two-Joint Cable-Driven Space and Joint Space

For the hyper-redundant robotic structure studied in this paper, the cable-driven device is placed in the base drive control module located at the root joint of the robot. Therefore, for the cable group in the n-th joint, its cable group passes through all the previous joints, which leads to coupling effects between the joints. Any movement of the joints before the n-th joint will affect the total length of the cable group in the n-th joint.

According to the serial structure of the robot, the joints are connected in series. When considering the space of two joints, the cables of Joint 2 must pass through the two routing discs of Joint 1 before they can be connected to the second routing disc of Joint 2. Any arbitrary motion of the two degrees of freedom of Joint 1 will affect the cable group length of Joint 2 at Joint 1, thereby affecting the overall cable length of the second group, thus forming a coupling relationship. Although the movement of Joint 1 affects the cable group length of Joint 2, the arbitrary movement of the two degrees of freedom of Joint 2 does not affect the cable group length of Joint 1. Therefore, this paper only studies the decoupling analysis of Joint 2 with respect to Joint 1.

Since the robot achieves motion through cable control, this study adopts the variation of the cable length, i.e., the change in cable length from the first routing disc of Joint 1 to the second routing disc of Joint 2 relative to the initial state.

Let the two-degree-of-freedom joint angles of Joint 1 and Joint 2 be ψ_1, α_1 and ψ_2, α_2, and the positions of the cables in Joint 1 and Joint 2 on the routing discs be β_1 and β_2, respectively. The cable length variation of Joint 2 mainly consists of two parts: (1) The cable variation caused by the rotation of the two joint angles ψ_2, α_2 of Joint 2 itself; (2) The cable variation of Joint 2 due to the coupling effect caused by the rotation of the two joint angles ψ_1, α_1 of Joint 1.

For the cable variation caused by the rotation of the two joint angles ψ_2, α_2 of Joint 2 itself, the variation of the three cables is given by Eq. (9):

$$\begin{cases} \Delta l_{2\_1} = f(\psi_2, \alpha_2, \beta_2) - d \\ \Delta l_{2\_2} = f(\psi_2, \alpha_2, \beta_2 + 120^\circ) - d \\ \Delta l_{2\_3} = f(\psi_2, \alpha_2, \beta_2 + 240^\circ) - d \end{cases} \tag{9}$$

where, Δl_{i_j} represents the variation of the j-th cable in the i-th joint, and $f(\psi, \alpha, \beta)$ is the cable length calculation formula.

For the cable variation of Joint 2 due to the coupling effect caused by the rotation of the two joint angles ψ_1, α_1 of Joint 1, when Joint 1 moves, the cable length of Joint 2 passing through Joint 1 also changes with ψ_1, α_1. Therefore, to eliminate the coupling effect caused by the motion of Joint 1, the cable length variation of Joint 2 needs to include the variation caused by the coupling effect, which is given by Eq. (10):

$$\begin{cases} \Delta l_{2\_1}' = f(\psi_1, \alpha_1, \beta_2) - d \\ \Delta l_{2\_2}' = f(\psi_1, \alpha_1, \beta_2 + 120^\circ) - d \\ \Delta l_{2\_3}' = f(\psi_1, \alpha_1, \beta_2 + 240^\circ) - d \end{cases} \tag{10}$$

where, $\Delta l_{i\_j}'$ represents the variation of the j-th cable in the i-th joint due to the coupling effect.

Thus, the total cable length variation of Joint 2 is given by Eq. (11):

$$\begin{cases} \Delta l_{21} = \Delta l_{2\_1} + \Delta l_{2\_1}' = f(\psi_2, \alpha_2, \beta_2) + f(\psi_1, \alpha_1, \beta_2) - 2d \\ \Delta l_{22} = \Delta l_{2\_2} + \Delta l_{2\_2}' = f(\psi_2, \alpha_2, \beta_2 + 120^\circ) + f(\psi_1, \alpha_1, \beta_2 + 120^\circ) - 2d \\ \Delta l_{23} = \Delta l_{2\_3} + \Delta l_{2\_3}' = f(\psi_2, \alpha_2, \beta_2 + 240^\circ) + f(\psi_1, \alpha_1, \beta_2 + 240^\circ) - 2d \end{cases} \tag{11}$$

where, Δl_{ij} represents the total variation of the j-th cable in the i-th joint.

2.3 Multi-joint Cable-Driven Space and Joint Space Decoupling Analysis

The super-redundant robot is mainly composed of 8 serially connected joints, and its coupling relationship is more complex. The decoupling analysis between the cable-driven space and joint space of the two joints obtained earlier is used to simplify the overall decoupling analysis of the robot. In this paper, the method mentioned above is used, and an iterative approach is applied to solve the total length of the cable groups for each joint.

Let the two degrees of freedom rotation angles of joint n be ψ_n, α_n, and the total cable length change of joint n be $\Delta l_{ni}(i = 1, 2, 3)$, where the position of the cable at the first joint's wiring disk is β_n. The expression for Δl_{ni} is given by Eq. (12):

$$\begin{cases} \Delta l_{n1} = \sum_{k=1}^{n} \left(f\left(\psi_k, \alpha_k, \beta_n\right)\right) - nd \\ \Delta l_{n2} = \sum_{k=1}^{n} \left(f\left(\psi_k, \alpha_k, \beta_n + 120°\right)\right) - nd \\ \Delta l_{n3} = \sum_{k=1}^{n} \left(f\left(\psi_k, \alpha_k, \beta_n + 240°\right)\right) - nd \end{cases} \tag{12}$$

The overall inverse kinematics analysis is based on the known joint angle sets $\psi_1, \alpha_1, \ldots, \psi_8, \alpha_8$, and the total cable length change for each joint, $\Delta l_{i1}, \Delta l_{i2}, \Delta l_{i3}(i = 1, 2 \ldots 7, 8)$, is calculated using Eq. (12).

3 BiLSTM-Based Forward Kinematics Solution Network Design

Let the 16 joint angles of the 8 universal joints be $\boldsymbol{\theta}^{(t)} = [\theta_1, \theta_2, \ldots, \theta_{16}]$; the 24 cable length change amounts on the 8 joints be $\boldsymbol{\Delta l}^{(t)} = [\Delta l_1, \Delta l_2, \ldots, \Delta l_8]$, where Δl_{i} is the 3 cable length changes at the i-th joint $[\Delta l_{\mathrm{i}1}, \Delta l_{\mathrm{i}2}, \Delta l_{\mathrm{i}3}]$. The forward kinematics solution network structure, training process, and performance evaluation are introduced below.

3.1 Basic Network Structure Design

A single residual fully connected module is shown in Figure 2. A short connection mechanism is used to solve the vanishing gradient problem in deep network training. In each residual module, the input passes through two fully connected layers and is added to the output through a short connection. By using the short connection and addition operation, the network can learn the difference, i.e., the residual, between the input and output, thus training and fitting more effectively. Finally, the residual modules are stacked to form the fully connected network.

For the BiLSTM-based network, the design goal is to analyze the sequential relationships between joint coordinate points. The BiLSTM network is based on the LSTM architecture, as shown in Figure 3. Each BiLSTM network consists of two LSTM networks, where each LSTM network has two inputs: the current feature input and the hidden layer. Compared to the unidirectional LSTM network, the BiLSTM network can better encode the relevant information between the input elements, capturing the associations in the input sequence during both forward and backward propagation.

For the forward kinematics solution of joint space and cable-driven space, two main network structures have been attempted. The first one is a single-joint space and cable-driven space forward kinematics network based on the decoupling method, aiming to handle the forward kinematics solution of each joint separately and calculate all joint angles through decoupling. The second one is the overall joint space and cable-driven space forward kinematics network, aiming to process the forward kinematics solution of all joints simultaneously.

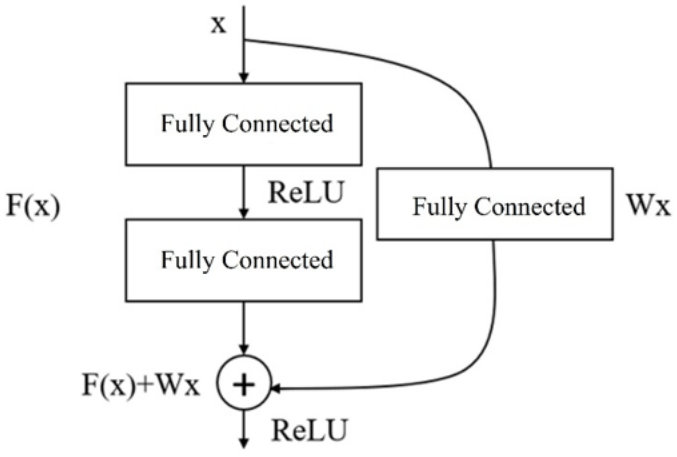

Figure 2. Single residual fully connected block

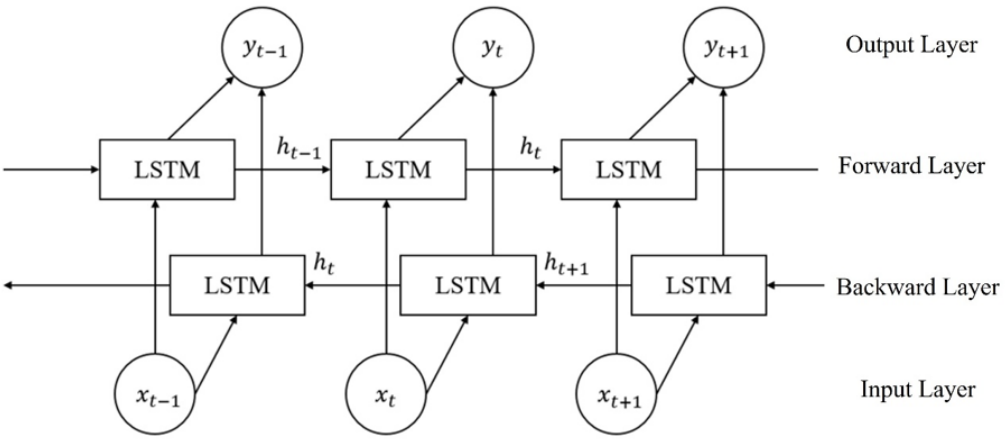

Figure 3. Basic BiLSTM network structure

3.2 Single-Joint Forward Kinematics Solution Network

First, a network based on BiLSTM combined with the decoupling method for single-joint forward kinematics solution is proposed, called the Single Joint Kinematic Solution Network based on BiLSTM (SJ-BiLSTM). The network uses a structure that combines BiLSTM layers and fully connected layers. The network input is the position β_n of the first cable on the disk and the three cable length changes $\Delta l_{ni}(i = 1, 2, 3)$ at the joint, while the network output is the two joint angles ψ_n and α_n of the two degrees of freedom of the joint. Figure 4 shows the network structure of the single-joint space and cable space, and the network structure parameters are shown in Table 1.

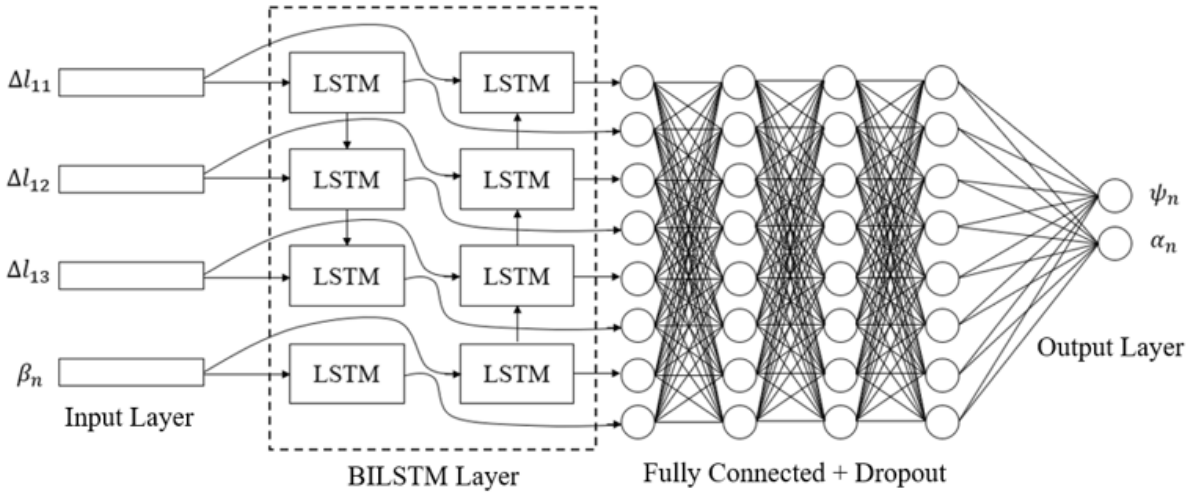

Figure 4. Single-joint forward kinematics network structure

Table 1. Single-joint forward kinematics network structure parameters

Network Layer Name	Network Output Dimension	Number of Layers in Module
Input Layer	4	-
BiLSTM Layer	2× Batch Size	-
Residual Fully Connected Module 1	2× Batch Size	2
Residual Fully Connected Module 2	2× Batch Size	2
Residual Fully Connected Module 3	2× Batch Size	2
Fully Connected Layer 1	2	-

After training the single-joint forward kinematics network, the decoupling method is used to solve the 16 joint angles $\boldsymbol{\theta}^{(t)}$ of the overall super-redundant robot. The specific method is as follows: Starting from the base, the joint angles ψ_i and $\alpha_i(\mathrm{i} = 1, 2, \ldots, \mathrm{n})$ of each joint are solved in sequence according to the joint order. When solving the

two joint angles ψ_i and α_i of joint i, the total cable length change of joints 1 to $i-1$ is calculated based on the joint angles of the previous $i-1$ joints, and then subtracted from the current cable change of joint i. This gives the cable length change caused by the two degrees of freedom of joint angle at joint i. This value, along with the first cable position β_i, is input into the single-joint forward kinematics solution network to calculate ψ_n and α_n.

3.3 Multi-Joint Forward Kinematics Solution Network

Furthermore, a network directly based on BiLSTM for multi-joint forward kinematics solution is proposed, called the Multiple Joints Kinematic Solution Network based on BiLSTM (MJ-BiLSTM). The network input is the 3-channel 8 sets of cable length changes $\boldsymbol{\Delta l}^{(t)}$, while the network output is the 16 joint angles $\boldsymbol{\theta}^{(t)}$. The overall joint space and cable-driven space forward kinematics network is shown in Figure 5, and the network structure parameters are shown in Table 2.

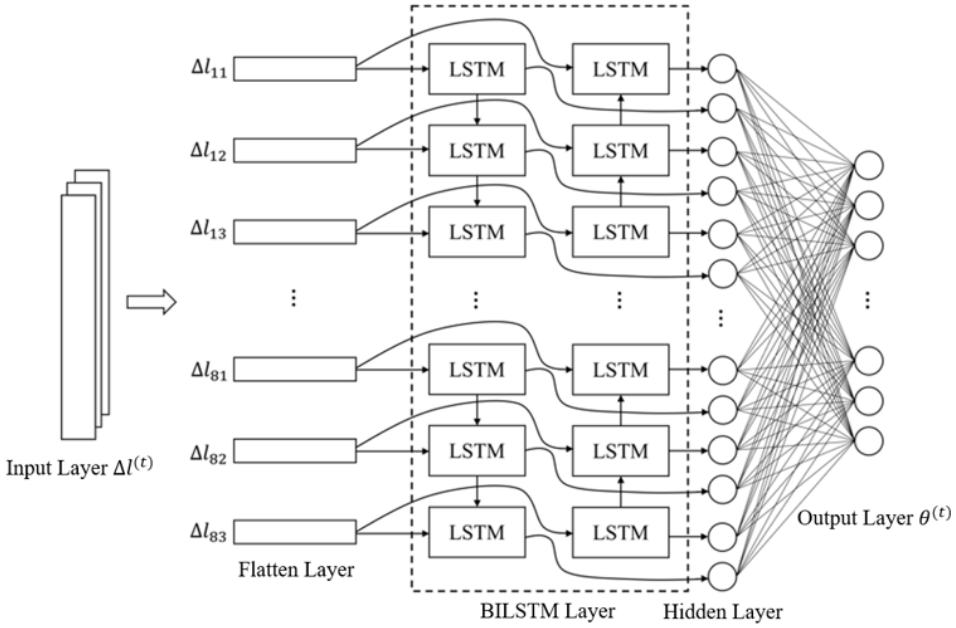

Figure 5. Multi-joint forward kinematics solution network structure

Table 2. Multi-joint forward kinematics solution network structure parameters

Network Layer Name	Network Output Dimension
Input Layer	16
BiLSTM Layer	2× Batch Size
Fully Connected Layer 1	24

3.4 Deep Network Training Set Generation

Deep learning is a supervised learning method that requires the given features and corresponding expected target values as training samples. In the training of the forward kinematics model for the superredundant robot, two sample sets need to be constructed. The first sample set is for the single-joint space and cable-driven space forward kinematics model. Through the mapping relationship derived in Section 2.2, and setting the two joint angles ψ_n and α_n to change within the range of -30° to 30° with a step size of 0.1°, and the first cable position β_n to change within the range of 0° to 119° with a step size of 1°, a dataset containing $43,344,240$ data points is generated. The second sample set is for the overall joint space and cable-driven space forward kinematics model. Through the mapping relationship derived in Section 2.3, and setting the 16 joint angles ψ_n and α_n to change within the range of -30° to 30° with a step size of 0.1°, along with the cable length changes, a dataset containing 50,000,000 data points is generated.

3.5 Deep Network Training Settings and Training Method

3.5.1 Deep network loss function settings

In deep network training, the loss function is used to measure the difference between the model's predicted result and the actual target. The loss function used in the deep network training in this paper is the Mean Squared Error (MSE). MSE calculates the squared differences between the predicted and actual values and finds the average of these squared differences. The formula for MSE is shown in Eq. (13):

$$\text{MSE} = \frac{\sum (y_i - \hat{y}_l)^2}{n} \tag{13}$$

where, n is the number of samples, y_i is the actual target value, and $\widehat{y}_i$ is the model's predicted value. The smaller the MSE loss function, the smaller the difference between the model's predicted result and the actual value, meaning the model's fitting effect is better. Therefore, the deep network model parameters can be adjusted to minimize the MSE loss function, gradually improving prediction accuracy.

3.5.2 Training data preprocessing

In deep learning, preprocessing the model's input is a common step [18]. This is because the dataset may have bias, which makes the network weights sensitive to changes and difficult to optimize the network parameters during gradient descent. The preprocessing method used in this paper is standardization, which converts the dataset's input to a distribution with a mean of zero, thereby improving the deep network's performance. The formula for standardization is shown in Eq. (14):

$$x' = \frac{x - \mu}{\sigma} \tag{14}$$

where, x is the original dataset input, x' is the standardized data, μ is the mean of the dataset input, and σ is the standard deviation of the dataset input. By calculating the difference between the input data and the mean, and dividing by the standard deviation, the input data can be standardized to a distribution with a mean of 0 and a standard deviation of 1. By making the input data have a uniform scale and distribution, the stability of gradient descent and network optimization can be improved, thus accelerating the network's convergence speed.

3.5.3 Learning rate decay

The learning rate in deep learning determines the step size for each parameter update. A proper learning rate helps the optimization algorithm effectively converge to a global optimal solution or local optimal solution. A learning rate that is too high may cause parameter updates to be too large, missing the optimal solution, while a learning rate that is too low may result in slow convergence and require more training time. Therefore, selecting a proper learning rate is an essential part of deep network training.

In learning rate settings, constant learning rate and variable learning rate are two common methods [19]. A constant learning rate remains unchanged throughout the training process, usually chosen based on experience. However, a constant learning rate cannot adapt to the different needs of training at various stages, which may lead to poor performance in the early or late stages of training.

In contrast, a variable learning rate is a more flexible method of setting the learning rate [20]. In this method, the learning rate gradually decreases as the network training progresses. By appropriately adjusting the size of the learning rate and its change strategy, the model's convergence speed and performance can be improved. This paper adopts the variable learning rate method, gradually lowering the learning rate, allowing the model parameters to converge more finely to the optimal position in the later stages of training. This approach helps the optimization algorithm better explore the parameter space, avoiding large parameter update steps and thus improving the model's convergence and generalization ability.

4 Experiment and Discussion

In order to analyze and compare the models, this paper uses the validation dataset for prediction and employs two commonly used evaluation metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). By calculating the difference between the predicted values and the actual values, the prediction accuracy is quantified, and the performance of the two models is compared.

Figure 6 and Figure 7 show the convergence of the two models, and it can be seen that both networks can effectively converge.

For the single-joint forward kinematics solving network, it can predict the change in the cable group length for a single joint and then use the traditional decoupling method to recursively solve for the robot's 16 joint angles. The MAE and RMSE of SJ-BiLSTM are shown in Figure 8, the joint angle errors solved by SJ-BiLSTM are shown in Figure 9.

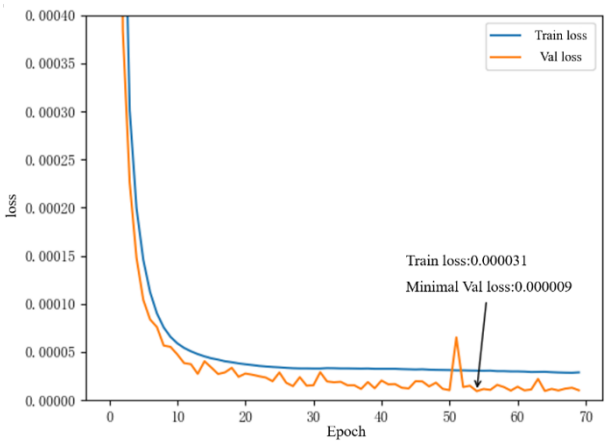

Figure 6. Convergence of SJ-BiLSTM

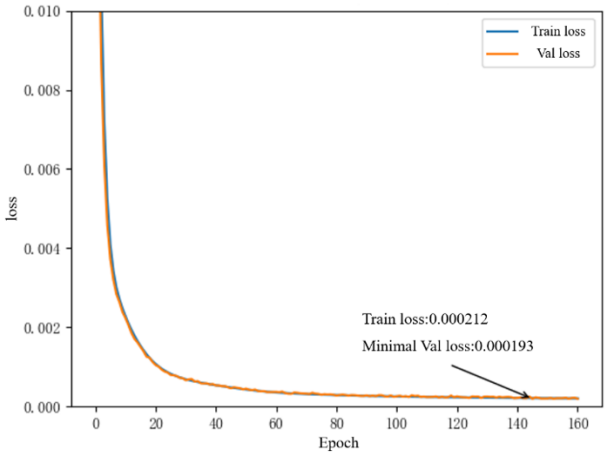

Figure 7. Convergence of MJ-BiLSTM

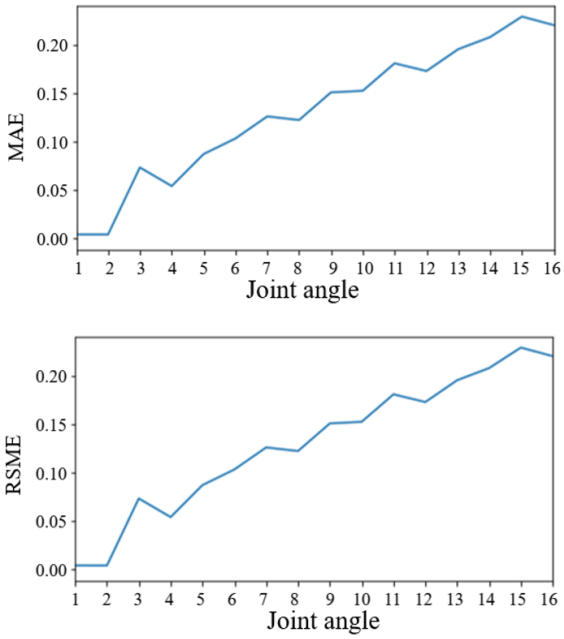

Figure 8. MAE and RMSE of SJ-BiLSTM

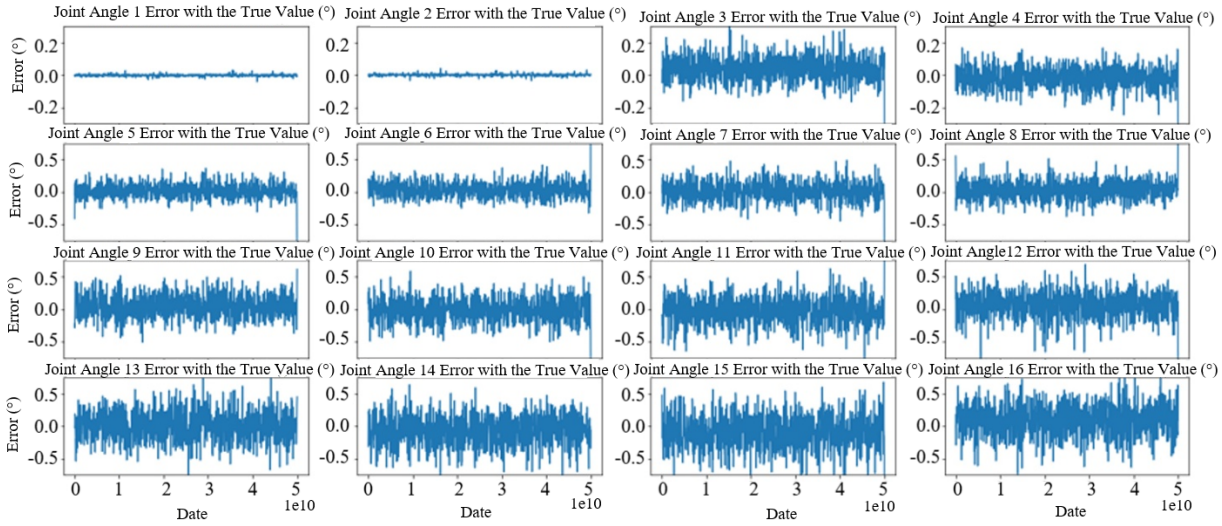

Figure 9. Joint angle errors solved by SJ-BiLSTM

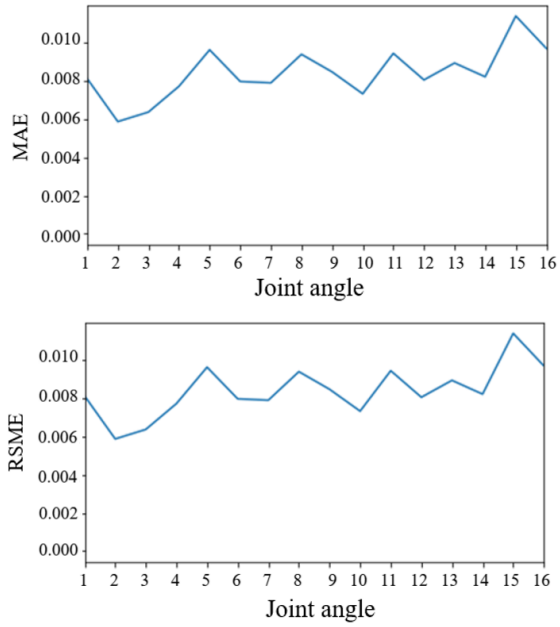

Figure 10. MAE and RMSE of MJ-BiLSTM

Table 3. Comparison of network performance

Model	MAE	RSME	Time (s)
SJ-BiLSTM	0.1295	0.1845	1151.6 s
MJ-BiLSTM	0.0084	0.0110	101.0 s

The multi-joint forward kinematics solving network directly predicts the 24 cable length change values as input. The MAE and RMSE of the joint angles are shown in Figure 10, and the cable length change errors are shown in Figure 11.

Comparing the average MAE and RMSE values and solving time of the two models, the results are shown in Table 3. It can be seen that the multi-joint forward kinematics solving network has smaller MAE and RMSE, which are an order of magnitude smaller than those of the single-joint forward kinematics solving network; and in terms of computation time, the multi-joint forward kinematics solving network does not require the decoupling method for recursively solving the robot joint angles, so its computational time is less than one-tenth of that of the single-joint

forward kinematics solving network.

Furthermore, the average value and standard deviation of the joint angle absolute errors solved by the two models are intuitively plotted on the same graph for comparison, as shown in Figure 12. It can be seen that due to the need for recursive solving in the single-joint forward kinematics solving network, the solving errors gradually accumulate and increase, while the multi-joint forward kinematics solving network does not have accumulated errors.

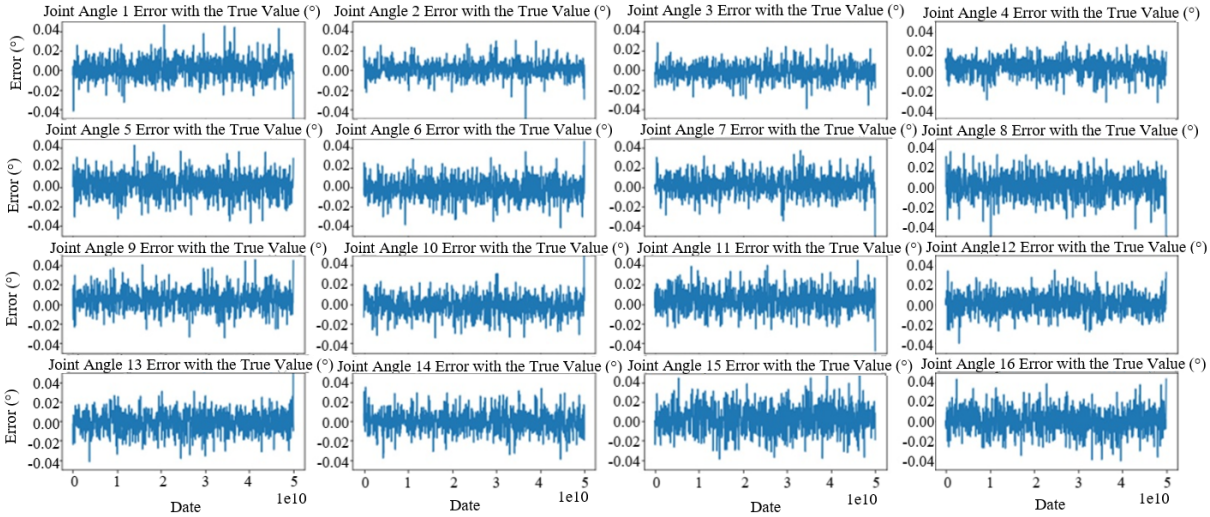

Figure 11. Joint angle errors solved by MJ-BiLSTM

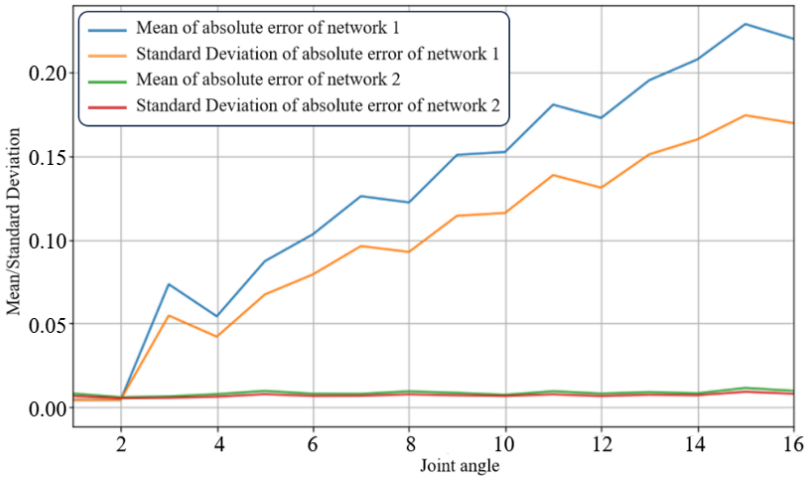

Figure 12. Distribution of joint angle absolute error mean and standard deviation for joint space and cable-driven space forward kinematics networks

5 Conclusions

This paper investigated the BiLSTM-based forward kinematics solving method for a 24-cable-driven ultra-redundant robotic arm. First, the single-joint kinematic model of the cable-driven ultra-redundant robotic arm was established, and the mapping relationship between cable length and joint angle was explored. Based on this, the coupling equations between cable length and joint angles are derived. Then, two network design ideas were proposed: one for solving single-joint forward kinematics and the other for solving multi-joint forward kinematics using BiLSTM, leading to the design of two network structures, SJ-BiLSTM and MJ-BiLSTM.

Through comparative experiments, we found that the multi-joint forward kinematics solving network not only outperforms the single-joint forward kinematics solving network in accuracy but also has a significant advantage in computational efficiency, with its computation time being only one-tenth that of the single-joint forward kinematics solving network.

The study verifies the feasibility and effectiveness of BiLSTM-based forward kinematics solving for cable-driven ultra-redundant robotic arms. It provides a new approach for kinematic modeling of cable-driven ultra-redundant robotic arms and serves as a reference for the application of deep learning in robotic kinematics problems. Future research can further optimize the network structure, improve the model's generalization ability, and extend its application in more complex environments.

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robot. Comput. Integr. Manuf.*, vol. 37, pp. 282–291, 2016. https://doi.org/10.1016/j.rcim.2015.04.002

[2] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu, "Three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1423–1434, 2017. https://doi.org/10.1109/TMECH.2017.2683561

[3] R. O. Buckingham and A. C. Graham, "Dexterous manipulators for nuclear inspection and maintenance — Case study," in *2010 1st International Conference on Applied Robotics for the Power Industry (CARPI)*, Montreal, QC, Canada, 2010, pp. 1–6. https://doi.org/10.1109/CARPI.2010.5624476

[4] Y. Yamauchi, Y. Ambe, H. Nagano, M. Konyo, Y. Bando, E. Ito, S. Arnold, K. Yamazaki, K. Itoyama, T. Okatani, H. G. Okuno, and S. Tadokoro, "Development of a continuum robot enhanced with distributed sensors for search and rescue," *Robomech J.*, vol. 9, no. 8, pp. 1–13, 2022. https://doi.org/10.1186/s40648-022-00223-x

[5] X. Dong, D. Axinte, D. Palmer, S. Cobos, M. Raffles, A. Rabani, and J. Kell, "Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines," *Robot. Comput. Integr. Manuf.*, vol. 44, pp. 218–229, 2017. https://doi.org/10.1016/j.rcim.2016.09.004

[6] R. Buckingham, V. Chitrakaran, R. Conkie, G. Ferguson, and A. Graham, "Snake-arm robots: A new approach to aircraft assembly," in *SAE Technical Paper*, 2007. https://doi.org/10.4271/2007-01-3870

[7] R. Bogue, "Robots in the nuclear industry: A review of technologies and applications," *Ind. Robot Int. J.*, vol. 38, no. 2, pp. 113–118, 2011. https://doi.org/10.1108/01439911111106327

[8] W. Hong, L. Xie, J. Liu, Y. Sun, K. Li, and H. Wang, "Development of a novel continuum robotic system for maxillary sinus surgery," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 3, pp. 1226–1237, 2018. https://doi.org/10.1109/TMECH.2018.2818442

[9] H. Yang, D. Meng, X. Wang, B. Liang, W. Xu, and P. Jiang, "Manipulability analysis for cable-driven hyper-redundant manipulators," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, Lyon, France, 2021, pp. 1480–1487. https://doi.org/10.1109/CASE49439.2021.9551676

[10] T. Liu, W. Xu, T. Yang, and Y. Li, "A hybrid active and passive cable-driven segmented redundant manipulator: Design, kinematics, and planning," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 2, pp. 930–942, 2021. https://doi.org/10.1109/TMECH.2020.3013658

[11] W. Xu, T. Liu, and Y. Li, "Kinematics, dynamics, and control of a cable-driven hyper-redundant manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 4, pp. 1693–1704, 2018. https://doi.org/10.1109/TMECH.2018.2842141

[12] A. Karpathy, J. Johnson, and F. F. Li, "Visualizing and understanding recurrent networks," *arXiv preprint*, 2015. https://doi.org/10.48550/arXiv.1506.02078

[13] Y. Chen, Z. Li, W. Xu, Y. Wang, and H. Ren, "Minimum sweeping area motion planning for flexible serpentine surgical manipulator with kinematic constraints," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015, pp. 6348–6353. https://doi.org/10.1109/IROS.2015.7354284

[14] H. Takatani, N. Araki, T. Sato, and Y. Konishi, "Neural network-based construction of inverse kinematics model for serial redundant manipulators," *Artif. Life Robot.*, vol. 24, pp. 487–493, 2019. https://doi.org/10.1007/s10015-019-00552-y

[15] H. Dong, L. Chen, W. T. Wu, L. G. Yao, and H. Sun, "A novel algorithm by combining nonlinear workspace partition with neural networks for solving the inverse kinematics problem of redundant manipulators," *Mech. Sci.*, vol. 12, no. 1, pp. 259–267, 2021. https://doi.org/10.5194/ms-12-259-2021

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. http://doi.org/10.1162/neco.1997.9.8.1735

[18] H. Y. Yang, X. Y. Zhao, and L. Wang, "Review of data normalization methods," *Comput. Eng. Appl.*, vol. 59, no. 3, pp. 13–22, 2023. https://doi.org/10.3778/j.issn.1002-8331.2207-0179

[19] L. Li, Z. J. Qiao, and Z. Q. Long, "A smoothing algorithm with constant learning rate for training two kinds of fuzzy neural networks and its convergence," *Neural Process. Lett.*, vol. 51, pp. 1093–1109, 2020. https://doi.org/10.1007/s11063-019-10135-4

[20] H. Iiduka, "Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13 250–13 261, 2022. https://doi.org/10.1109/TCYB.2021.3107415