Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **1**, February **2025**

Intelligent Trajectory Control of Sampling Robotic Arm on Bidirectional LSTM Neural Network

Jin-Yu Xu

School of Electrical and Information Engineering
Hubei University of Automotive Technology, Shiyan 442002, P. R. China
xujinyuhuat@163.com

Ze-Ya Li

School of Electrical and Information Engineering
Hubei University of Automotive Technology, Shiyan 442002, P. R. China
li_zeya@163.com

Zhou-Rong Li*

College of Electronic Information
Chongqing Institute of Engineering, Chongqin 400056, P. R. China
cqie_li@163.com

Clement Li

College of Information Technology
Waikato Institute of Technology, Hamilton 3240, New Zealand
gc0744@163.com

*Corresponding author: Zhou-Rong Li

Received January 31, 2024, revised May 6, 2024, accepted June 21, 2024.

Abstract. *A sampling robotic arm usually consists of multiple joints, allowing the arm to perform various motions and actions in three-dimensional space. Aiming at the problems of low sampling accuracy and poor motion smoothness in the work of the sampling robotic arm control system, this paper proposes an intelligent trajectory control method for the sampling robotic arm based on bidirectional LSTM neural network. Firstly, the kinematic model of the sampling robotic arm is established by analysing the interrelationships of the joints of the sampling robotic arm, and the kinematic analysis is completed by using the D-H method. Then, a Bidirectional Time-aware LSTM (Bi-T-LSTM) combined with Self-encoder is proposed. A bi-directional autoencoder is constructed using the time-aware LSTM unit, which combines the forward T-LSTM and the backward T-LSTM, in order to better capture the bi-directional trajectory information. Finally, a Self-encoder Bi-T-LSTM neural network controller is designed based on the sampled robotic arm kinematic model. The neural network controller is used to virtually control the inputs to compensate for the unknown mechanical system dynamics. The sampling robotic arm is simulated using Matlab/Simulink Simscape-Multibody and Simscape-Fluids simulation platforms. The simulation results show that the proposed intelligent trajectory control algorithm can help the robotic arm to optimise the trajectory planning more efficiently, achieve good position control, and ensure that the system output does not exceed the preset range.*

Keywords: Sampling robotic arm; trajectory planning; kinematic analysis; Bi-LSTM; self-encoder

1. **Introduction.**

In the manufacturing industry, robotic arms are commonly used for assembly, fitting, handling and packaging. They can perform highly repetitive tasks to improve productivity and product quality [1, 2, 3]. Sampling robotic arms are usually equipped with various attachments and tools such as grippers, suction cups, etc. to be able to adapt to different sampling needs. By using sampling robotic arms, automated sample collection, transfer and analysis can be achieved, increasing efficiency and reducing human error [4]. Sampling robotic arms can also play an important role in sampling tasks that are hazardous or require a high degree of accuracy, such as coal sampling [5, 6].

The kinematics of the sampling manipulator mainly studies the position and posture of the sampling head of the sampling manipulator in its workspace, the relationship between the joint variables and the motion relationship between the joints, which is the most basic research content of kinematics analysis [7]. The kinematics of the sampling robotic arm mainly involves trajectory planning, offline programming and the design of motion control algorithms.

Multi-robot collaboration refers to the nature of multiple robots working in concert with each other when completing the same task [8, 9]. For complex operational tasks, such as handling irregularly shaped or handling larger and heavier objects, when a single robot is unable to complete the handling task due to its own limitations, multiple robots can replace a single robot to complete it through collaborative work, which also improves the efficiency of the robotic system in the process of the operation and enables the multi-robot system to solve more practical application problems [10]. The sampling robotic arm must follow the principle that the movement process is as smooth and steady as possible, which means that sudden changes in position, velocity and acceleration should be avoided. If the movement is not smooth it will increase the wear of mechanical parts and lead to the vibration and impact of the coal sampling robotic arm, which will reduce the sampling accuracy [11]. Obviously, to make the sampling robotic arm have good working performance and achieve real-time control, the analysis and study of its kinematics is very important.

Intelligent trajectory control of a sampling robotic arm refers to the use of advanced control algorithms and technologies to enable the robotic arm to autonomously learn, plan, and execute complex motion trajectories in order to complete various sampling tasks [12]. Neural network is a computational model that imitates the neuronal network structure of the human brain, and is capable of achieving complex nonlinear mapping by learning a large amount of data for pattern recognition and decision making. In the intelligent trajectory control of robotic arms, complex control strategies can be realised through the nonlinear approximation ability of neural networks to adapt to different sampling tasks and environmental changes. Neural networks can learn and predict suitable trajectories of robotic arms in different workspaces to optimise path planning and avoid collisions [13, 14]. Therefore, the research objective of this work is to use neural network technology to realise intelligent trajectory control of sampling robotic arm, so as to enhance its operation ability in various complex environments.

1.1. **Related Work.**

Current research in robotic arm trajectory control focuses on improving the accuracy and efficiency of trajectory planning, enhancing kinematic modelling for robotic arm control, and exploring real-time control methods in complex environments [15, 16].

Seraji and Bon [17] proposed a method for real-time collision avoidance during robotic arm motion. This method can monitor obstacles during the movement of the robotic arm and adjust the trajectory in real time to ensure that the robotic arm completes the task safely and effectively. Ren et al. [18] studied the control of multi-degree-of-freedom robotic arm trajectory using PID plus weight-weighted force-compensated control method, and performed experimental analyses and validation in the aspects of robotic arm trajectory tracking, anti-interference ability, etc. Ramabalan et al. [19] proposed a dynamic path planning method for robotic arms based on dynamic programming. The method considers both kinematic and dynamic constraints, and can effectively plan smooth, continuous and dynamically feasible path trajectories. Elbanhawi et al. [20] proposed to use B-spline curves to represent and model the dynamic path trajectories of the robotic arm, and gave a detailed algorithmic process. Liu et al. [21] used genetic algorithms to find the beginning and end points under the constraints of speed, acceleration and speed of the robotic arm.

Artificial intelligence technology can provide more powerful learning and modelling capabilities, which can help to achieve better trajectory control of robotic arms. On the basis of traditional optimization algorithms such as genetic algorithm or particle swarm optimization, neural network modelling techniques are added to achieve deep exploration of the solution space and better path solutions. Giuffrida et al. [22] proposed a Convolutional Neural Network (CNN)-based image-driven control of the robotic arm, which can learn the joint space from the image inputs directly to achieve better end-to-end control results. Liu et al. [23] used deep reinforcement learning methods to train the robotic arm to learn the specified operation. Liu et al. [23] used a deep reinforcement learning method to train a robotic arm to learn a specified sequence of operational actions, and achieved powerful trial-and-error based robotic arm action learning. Shi et al. [24] proposed to use a Recurrent Neural Network (RNN) structure to represent and learn the robotic arm motion patterns, and achieved good continuous path control results. However, the RNN's long-range dependence on previous information makes its learning ability degrade over time, leading to serious discrepancies between the training results and the desired goals.

1.2. **Motivation and contribution.** Bidirectional LSTM (Bi-LSTM) [25] is a recurrent neural network model that can effectively capture long-term dependencies in sequential data, and has both forward and backward memory units, which can simultaneously learn contextual information in a sequence. However, ordinary RNNs can only learn forward context. This can improve the control accuracy by making full use of the context information for robotic arm motion trajectory learning. Bi-LSTM has a larger number of parameters than RNN and has a better learning ability. It can better model the nonlinear characteristics in robotic arm motion. Therefore, in order to effectively improve the control accuracy of the sampling robotic arm, a Bi-LSTM-based robotic arm trajectory control method is proposed. The main innovations and contributions of this paper include:

(1) The implicated motion relationship between each linkage of the sampling robotic arm is studied, and then the linkage coordinate system of the neighbouring rods is established for the sampling robotic arm, and the kinematic equations of the sampling robotic arm are further established by using the D-H transformation, and the positive kinematic analysis is completed.

(2) In order to further explore the intrinsic connection between the current record and the records of the past and future moments, a Self-encoder Bi-T-LSTM is proposed, which can deepen the level of feature extraction for the original sequences to improve the precision of the trajectory control of the robotic arm.

(3) A robotic arm controller based on Self-encoder Bi-T-LSTM neural network was designed. The neural network controller was used to virtually control the input to compensate for the unknown mechanical system dynamics, and the corresponding neural network weight control law was designed.

2. **Kinematic analysis of the sampling robotic arm.**

2.1. **Position and attitude.** In order to describe the position of the end-effector of the sampling robotic arm under the workspace coordinate system, a coordinate system is selected as the reference coordinate system, as shown in Figure 1, and the position vector p is used to represent this attitude.

$$p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{1}$$

where p_x, p_y, p_z denote the coordinate components of the end-effector in the coordinate system; T matrices with this superscript are transposed matrices.

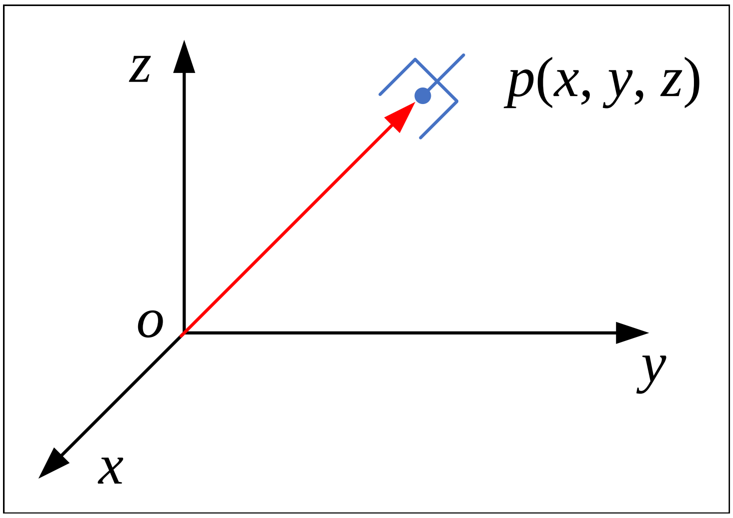


Figure 1. Spatial coordinates of point p

Let the coordinate system $\{i\}$ and the coordinate system $\{j\}$ have the same orientation, i.e., the direction vectors of the corresponding coordinate axes are parallel and in the same direction, but the coordinate origins of the two do not coincide. If the origin of the coordinate system $\{i\}$ and the coordinate system $\{j\}$ are in the same coordinate space, and the two are represented by the vector $\vec{P}_{ij}$, then the coordinate system $\{j\}$ can be transformed by translating the coordinate system $\{j\}$ along the vector $\vec{P}_{ij}$. The coordinate system $\{i\}$ can be obtained from the transformation result. The translation transform matrix is a 3×1 column matrix.

$$\vec{P}_{ij} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{2}$$

If there is a point in the space which can be represented by vectors $\vec{r}_i$ and $\vec{r}_j$ in the coordinate system $\{i\}$ and coordinate system $\{j\}$ respectively, then according to the principle of superposition transformation of vectors, the relationship between them is $\vec{r}_i = \vec{P}_{ij} + \vec{r}_j$ and we call this equation as the coordinate translation equation.

In order to describe the attitude of the end-effector of the sampling robot arm in the spatial coordinate system, as shown in Figure 2, a reference coordinate system $Oxyz$ is chosen. Another coordinate system $O_hX_hY_hZ_h$ is set at the centre of the reference coordinate system, and the 3×3 order matrix consisting of the cosines of the angles of the three directions with respect to the attitude in the reference coordinate system can be expressed by R.

$$R = \begin{bmatrix} \cos(x, x_h) & \cos(x, y_h) & \cos(x, z_h) \\ \cos(y, x_h) & \cos(y, y_h) & \cos(y, z_h) \\ \cos(z, x_h) & \cos(z, y_h) & \cos(z, z_h) \end{bmatrix} \tag{3}$$

where R is the rotation transformation matrix.

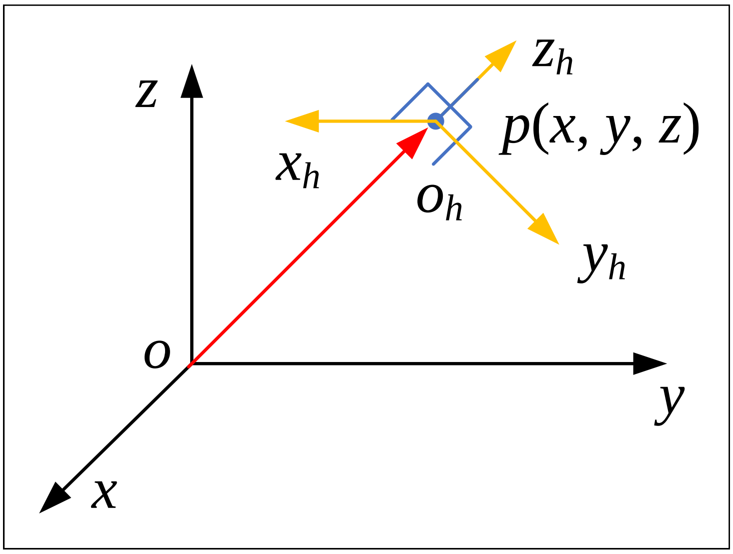


Figure 2. Posture description

Let the coordinate system $\{i\}$ and the coordinate system $\{j\}$ share the same coordinate system origin, but they have different orientations. The coordinate system $\{j\}$ can be obtained by rotating the coordinate system $\{i\}$ around the rotational transformation matrix. Compared with the translational transformation matrix, the rotational transformation matrix is designed with more parameters, among which the rotation around a single coordinate axis is the most basic form.

The transformation matrix is derived by rotating the θ angle around the z axis. Suppose there is any point p in space, then it can be represented by a coordinate component in the coordinate system $\{i\}$. It can also be represented by a coordinate component in the coordinate system $\{j\}$, but the two coordinate components are different. The relationship between the two is shown below.

$$\begin{cases} x_i = x_j \cdot \cos\theta - y_j \cdot \sin\theta \\ y_i = x_j \cdot \sin\theta + y_j \cdot \cos\theta \\ z_i = z_j \end{cases} \tag{4}$$

By completing the corresponding terms, the relational equation is then deformed by homogenizing the structural form.

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} \tag{5}$$

Expressing Equation (5) in vector form, the coordinate rotation transformation equation is obtained as follows:

$$\vec{r}_i = R_{ij}^{z,\theta} \cdot \vec{r}_j \tag{6}$$

where $\vec{r}_i$ is the coordinate vector of the p point in the coordinate system $\{i\}$; and $R_{ij}^{z,\theta}$ is the rotation transformation matrix.

The rotation transformation matrix for rotating the α angle around the x axis is:

$$R_{ij}^{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{7}$$

The rotation transformation matrix for rotating the β angle around the y axis is:

$$R_{ij}^{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{8}$$

2.2. **Chi-Square coordinate transformation.** In order to study the influence of the motion of each joint on the position and attitude of the end-effector of the sampling robotic arm, this paper adopts the chi-square coordinate system transformation, because the method of chi-square coordinate system transformation can be a good solution to the problem of describing the transformations of the position and attitude of the sampling robotic arm.

In a given coordinate space, three coordinate components are randomly given in (x, y, z). Determine a point using these fixed points (the three coordinate components). Relate the three right-angle coordinate components of this point to four numbers (x', y', z', k) that are not simultaneously zero as follows.

$$\begin{cases} x = \frac{x'}{k} \\ y = \frac{y'}{k} \\ z = \frac{z'}{k} \end{cases} \tag{9}$$

where (x', y', z', k) is the chi-square coordinate of the point in space [26].

The transformation matrix for the sum of vectors at any point in space in coordinate system $\{i\}$ and coordinate system $\{j\}$ is shown below:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} \tag{10}$$

Equation (10) is then expressed in terms of the coordinate component equations:

$$\begin{cases} x_i = p_x + \cos\theta \cdot x_j - \sin\theta \cdot y_j \\ y_i = p_y + \sin\theta \cdot x_j + \cos\theta \cdot y_j \\ z_i = p_z + z_j \end{cases} \tag{11}$$

Based on the above-mentioned chi-square coordinates, the transformations are processed by filling in the missing terms (the missing terms are replaced by zeros), and then the deformation of piecing together the corresponding terms is performed.

$$\begin{cases} x_i = \cos\theta \cdot x_j - \sin\theta \cdot y_j + 0 \cdot z_j + p_x \cdot 1 \\ y_i = \sin\theta \cdot x_j + \cos\theta \cdot y_j + 0 \cdot z_j + p_y \cdot 1 \\ z_i = 0 \cdot x_j + 0 \cdot y_j + 1 \cdot z_j + p_z \cdot 1 \\ 1 = 0 \cdot x_j + 0 \cdot y_j + 0 \cdot z_j + 1 \cdot 1 \end{cases} \tag{12}$$

$$M_{ij} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & p_x \\ \sin\theta & \cos\theta & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{ij}^{z,\theta} & \vec{p}_{ij} \\ 0 & 1 \end{bmatrix} \tag{13}$$

In the above chi-square transformation matrix, the 3 × 3 sub-matrix in the upper left corner is the rotation transformation matrix between the dynamic coordinate system and the static coordinate system, which clearly describes the positional relationship between the different coordinate systems. The 3 × 1 sub-matrix in the upper right corner of this chi-square coordinate transformation matrix is the matrix representing the translational transformation relationship between the two coordinate systems mentioned above, which describes the positional relationship between the two. So the chi-square coordinate transformation matrix is also called the position transformation matrix.

2.3. **Kinematic modelling of the sampling robotic arm.** D-H (Denavit-Hartenberg matrix) [27] was used to establish the rod coordinate system for the sampling robotic arm and derive the kinematic equations.

As shown in Figure 3, there are two joints at each end of the link, joint i and joint $i-1$. The dimensions of the rod can be described by two parameters, one is the rod length and the rod torsion angle, where the rod length can be expressed by the axes of each of the two neighbouring joints l_i, and the rod torsion angle can be expressed by the angle between the axes of each of the two joints of the rod at each end α_i. Finally, consider the relationship between a connecting rod i and a neighbouring rod $i-1$.

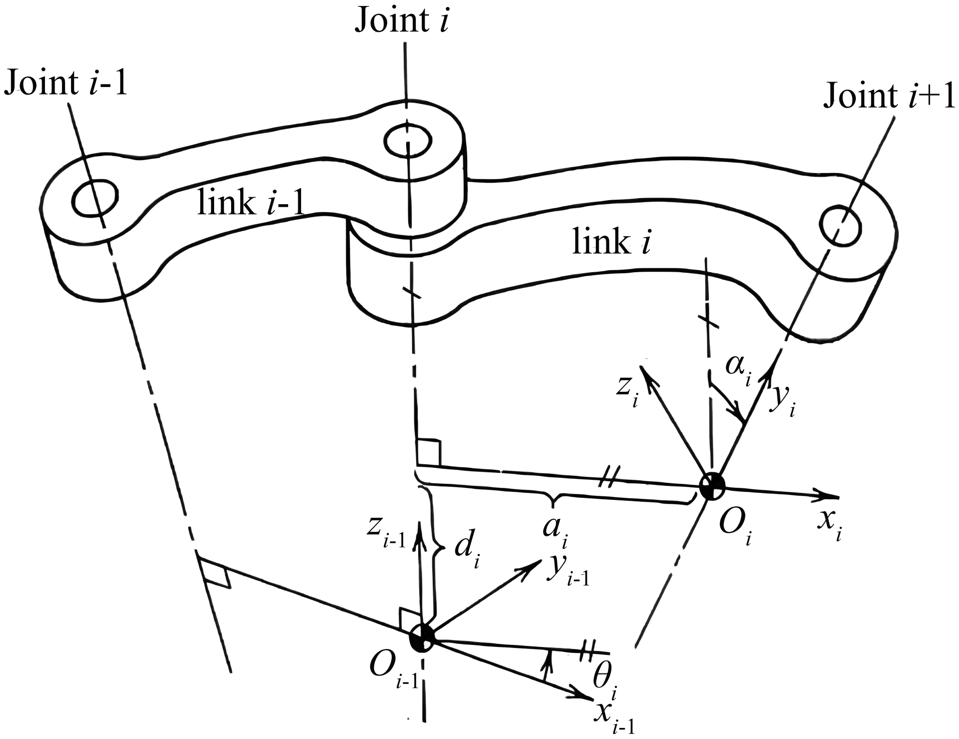


Figure 3. Rod joint kinematic parameters

After knowing the individual linkage parameters of the industrial robot, the structural dimensions of the sampled robotic arm can be analysed to obtain its linkage parameters as shown in Table 1.

Table 1. Sampling arm linkage parameters

Connecting rod	**Cornering variable θ_n**	**Connecting rod spacing d_n**	**Connecting rod length l_n**	**Connecting rod torsion angle α_n**
Connecting rod 1	θ_1	$d_1 = 0$	l_1	$\alpha_1 = 0$
Connecting rod 2	θ_2	$d_2 = 0$	l_2	$\alpha_2 = 0$
Connecting rod 3	θ_3	$d_3 = 0$	l_3	$\alpha_3 = 0$
Connecting rod 4	θ_4	$d_4 = 0$	l_4	$\alpha_4 = 0$

The angle variable θ_1 for the large arm (connecting rod 1) has a range of $[16°, 97°]$. The angle variable θ_2 for the small arm (rod 2) varies from $[-34°, 60°]$. The link length l_3 of the telescopic arm (link 3) varies in the range $[436, 1986]$. The range of variation of the sampling head (linkage 4) is $[-50°, 26°]$. The above data are obtained from the design parameters in the engineering design manual of the coal sampling robotic arm.

The big arm, the small arm, the telescopic arm, and the sampling head of the sampling robotic arm each have one degree of freedom of motion. According to Equation (14) and the kinematic parameters of the sampling robotic arm, the transformed relationship between the coordinate system of the base and the linkage coordinate system of the end-effector, i.e., the kinematic equation of the sampling robotic arm, can be obtained. The kinematic equations of the sampling robotic arm are calculated as shown below:

$$T_4 = A_1 A_2 A_3 A_4 \tag{14}$$

$$\begin{cases} A_1 = Rot(X_0, q_1)Trans(0, l_1, 0) \\ A_2 = Rot(X_1, q_2)Trans(0, l_2, 0) \\ A_3 = Trans(0, l_3, 0) \\ A_4 = Rot(X_4, q_4)Trans(0, l_4, 0) \end{cases} \tag{15}$$

$$Rot(X_0, \theta_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 & 0 \\ 0 & \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$Trans(0, l_1, 0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

On the basis of the geometric parameters and joint variables of each linkage of the sampling robotic arm, the kinematic equations can be obtained through Equation (14), and this process is considered to be a positive solution of the kinematics, i.e., it is the position of the coordinate system of the end-effector.

3. Bi-LSTM-based trajectory control of a sampling robotic arm.

3.1. **Principle of Bi-LSTM networks.** The hidden layer state of a classical LSTM network is only determined by the input information before a certain moment, and thus can only learn the input data before that moment. The Bi-LSTM network, which introduces a two-way propagation mechanism on the basis of the traditional LSTM network, can well combine the sequence data before and after a certain moment to make relevant predictions.

The Bi-LSTM network adds a new backpropagation layer, and the state of the neurons in the backpropagation layer is determined by the input data after a certain moment. The output of the network at that moment is determined by the inputs at that moment, the state value of the forward hidden layer and the state value of the backward hidden layer. In Bi-LSTM, x_t denotes the time-series data of the inputs at moment t, and h_t and h'_t denote the forward and backpropagation hidden states, respectively, with the following expressions.

$$\begin{cases} h_t = f(W_t h_{t-1} + U_t x_t + b_t) \\ h'_t = f(W'_t h'_{t+1} + U'_t x_t + b'_t) \end{cases} \tag{18}$$

where W_t, U_t, and b_t correspond to the weight value and bias term of the forward propagation hidden state h_t, respectively, and W'_t, U'_t, and b'_t correspond to the weight value and bias term of the backward propagation hidden state h'_t. The function f is the corresponding activation function.

The expression for the output y_t of the network is as follows:

$$y_t = g(Vh_t + V'h'_t + b_y) \tag{19}$$

where V and V' are the weight values of the forward-propagating hidden state h_t and the backward-propagating hidden state h'_t, b_y is the bias term, and g is the activation function, respectively.

For the trajectory control problem, the Bi-LSTM network incorporates the two structures of forward propagation layer and backpropagation layer. Compared with the traditional LSTM network, the Bi-LSTM network is able to comprehensively consider the robotic arm trajectory data before and after the moment to be controlled, and has a better grasp of the data as a whole.

3.2. **Time-aware Bi-LSTM combined with self-encoder.** Time intervals in robotic arm trajectory sequence data are not always uniformly distributed. Different time intervals can be considered as part of the information contained in the sequence data. If the time intervals are too far apart, the current output has very little dependence on the previous record. In this case, the contribution of the memory of the previous unit to the current unit should be ignored. However, Bi-LSTM cannot meet the above requirements.

Time-aware LSTM (T-LSTM) [28] does not discard the long-term memory completely, but performs subspace decomposition for the memory of the previous time step. Therefore, this paper adopts T-LSTM to construct Bi-T-LSTM, and the internal structure of the T-LSTM unit is shown in Figure 4. Bi-T-LSTM can take into full consideration of the positive and negative information law of the temporal data, further excavate the intrinsic connection between the current record and the records of the past and the future moments, and deepen the level of the original sequence of feature extraction, so as to improve the precision of the trajectory control of the robotic arm.

Firstly, the long-term memory C_{t-1} of the previous unit is obtained through the network, and the short-term memory C^S_{t-1} is obtained using tanh as the activation function.

$$C^S_{t-1} = \tanh(W_d C_{t-1} + b_d) \tag{20}$$

where W_d, b_d is the network parameter for the subspace decomposition.

Use a non-increasing function $g(\cdot)$ to convert the interval time Δt into appropriate weights and multiply it with short-term memory to obtain the discounted short-term memory $\hat{C}^S_{t-1}$.

$$\hat{C}^S_{t-1} = C^S_{t-1} \times g(\Delta t) \tag{21}$$

Subtract the short-term memory C_{t-1} from the long-term memory C^S_{t-1} to obtain the complementary subspace C^T_{t-1} of the long-term memory, which is added to the discounted short-term memory $\hat{C}^S_{t-1}$ to form the adjusted memory of the previous unit C^*_{t-1}.

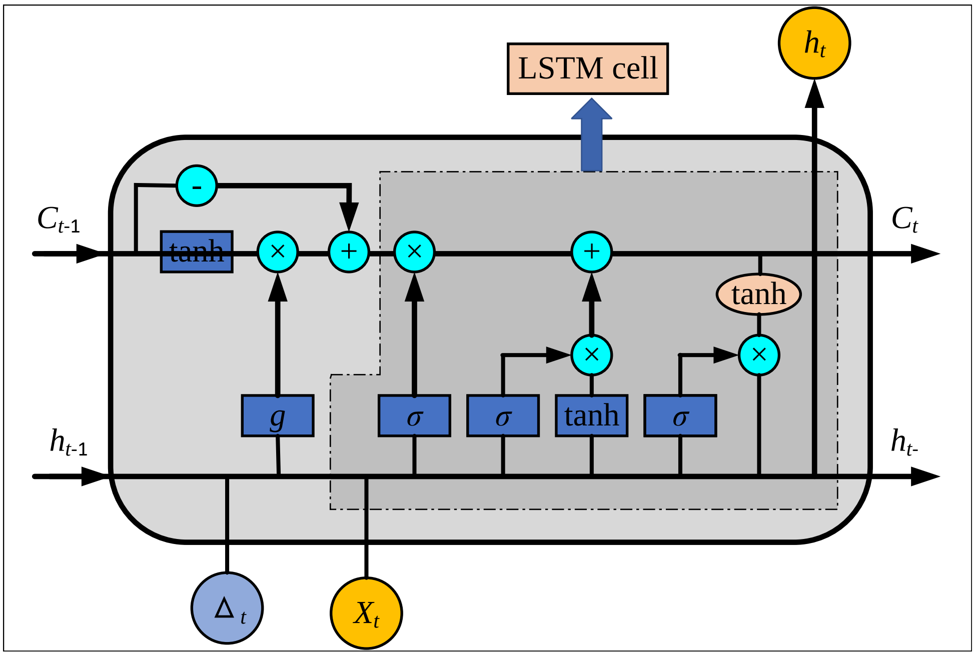


Figure 4. Internal structure of the T-LSTM cell

$$C_{t-1}^{T} = C_{t-1} - C_{t-1}^{S} \tag{22}$$

$$C_{t-1}^{*} = C_{t-1}^{T} + \hat{C}_{t-1}^{S} \tag{23}$$

Finally, the LSTM standard gating architecture is built. According to different application scenarios for the selection of different types of monotonically increasing functions. If the time interval between two consecutive records in the dataset is small, Equation (24) is used. If the value of the time interval is large, use Equation (25).

$$g(\Delta t) = \frac{1}{\Delta t} \tag{24}$$

$$g(\Delta t) = \frac{1}{\log(e + \Delta t)} \tag{25}$$

Bi-T-LSTM is able to splice the outputs of forward and backward T-LSTM. Compared with the unidirectional T-LSTM, it can obtain the information from both front-to-back and back-to-front directions at the same time. The output H of Bi-T-LSTM can be obtained by splicing the two outputs.

$$H = [h_L \oplus h_R] \tag{26}$$

Where, h_L denotes the output of forward-order T-LSTM hidden layer, and h_R denotes the output of reverse-order T-LSTM hidden layer. When processing time series data, more information can be obtained, which is beneficial to the subsequent tasks, so Bi-T-LSTM can obtain better results than unidirectional T-LSTM in solving time series data problems.

Although Bi-T-LSTM can fully consider the forward and backward information laws of temporal data, it requires a large amount of storage space to preserve the hidden state when dealing with long sequences. In contrast, the self-encoder can solve this problem by learning low-dimensional data representations and encoding high-dimensional input data into lower-dimensional hidden representations, thus greatly reducing the required storage space. In addition, another problem with Bi-T-LSTM is that it may be overly concerned with local features of the sequence data while ignoring the overall data structure. Self-encoders, on the other hand, can compensate for this by first learning the global data distribution and then reconstructing the inputs in the decoding phase, allowing the model to better capture and understand the global data features.

Therefore, in this paper, Bi-T-LSTM and self-encoder are combined, as shown in Figure 5, in order to utilise the advantages of both to further improve the accuracy and speed of trajectory control. The proposed auto-encoder model in Self-encoder Bi-T-LSTM mainly consists of an encoder and a decoder, complemented by the nonlinear feature extraction capability of deep neural network, which aims to convert inputs into intermediate variables and these variables into outputs.

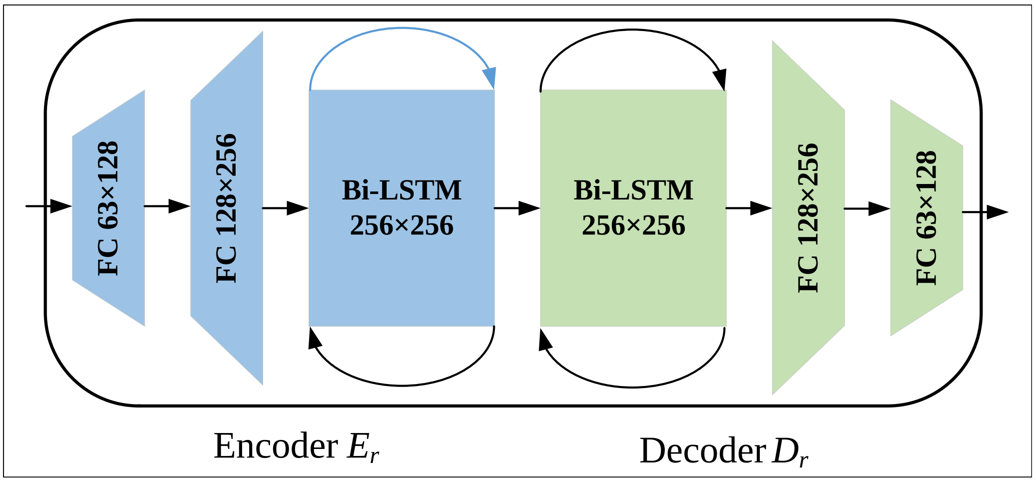


Figure 5. Self-encoder Bi-T-LSTM

3.3. **Controller design.** Based on the kinematic model of the sampled robotic arm, a Self-encoder Bi-T-LSTM neural network controller is designed. Take the joint space as an example, define $x_1 = q$, $x_2 = \dot{q}$, $x_3 = A_1P_1 - A_2P_2$. To facilitate the analysis, the sampling robotic arm dynamics model is written in state space form. The state error is defined as follows:

$$e_1 = x_1 - q_r, \quad e_2 = x_2 - \alpha_1, \quad e_3 = x_3 - \alpha_2^c \tag{27}$$

where α_1 is the virtual control input, α_2^c is the filtered signal of the virtual control input α_2.

Based on the generalised approximation performance of neural networks, we use Self-encoder Bi-T-LSTM neural network to virtually control the input α_2 to compensate for the unknown mechanical system dynamics. The designed neural network weight control law is shown below:

$$\widehat{W}_i = -\Gamma_i \left(\phi_i(Z) e_{2,i} + \sigma_i \widehat{W}_i \right) \tag{28}$$

where $Z = \left[q_r^T, \dot{q}_r^T, \ddot{q}_r^T, x_1^T, x_2^T\right]^T$ Γ_i is a positive definite symmetric matrix, and σ_i is a small positive constant.

4. **Simulation tests.**

4.1. **Experimental setup.** In this paper, a hydraulic robotic arm is set up by Matlab/Simulink Simscape-Multibody and Simscape-Fluids simulation platforms in order to verify the effectiveness of the adaptive neural network supple control strategy with output constraints. In order to facilitate the design of the controller, the displacement relationship between the joint space of the sampled robotic arm and the actuator space is set as $h_{a1}(q_1)$, $h_{a2}(q_2)$, $h_{a3}(q_3)$.

$$h_{a1} = q_1 \tag{29}$$

$$h_{a2} = \sqrt{L_{AB}^2 + L_{BC}^2 - 2L_{AB}L_{BC}\cos(q_2 + \theta_0 - \theta_1)} - h_{20} \tag{30}$$

$$h_{a3} = \sqrt{L_{DF}^2 + L_{EF}^2 - 2L_{DF}L_{EF}\cos(\pi - \gamma - q_3)} - h_{30}, \quad \gamma = \theta_2 + \theta_3 \tag{31}$$

4.2. **Positional tracking performance test.** The kinematic analysis and the Self-encoder Bi-T-LSTM based controller lead to an optimal solution for each joint variable, which uniquely determines one motion trajectory. The desired motion trajectory of each joint of the sampled robotic arm is shown below:

$$\theta_1 = 14.194 + 0.886t^2 - 0.749t^3, \quad t \in [0, 5] \tag{32}$$

$$\theta_2 = -3.483 - 0.613t^2 + 2.665t^3, \quad t \in [0, 5] \tag{33}$$

$$\theta_3 = -11.632 + 6.856t^2 - 2.021t^3, \quad t \in [5, 10] \tag{34}$$

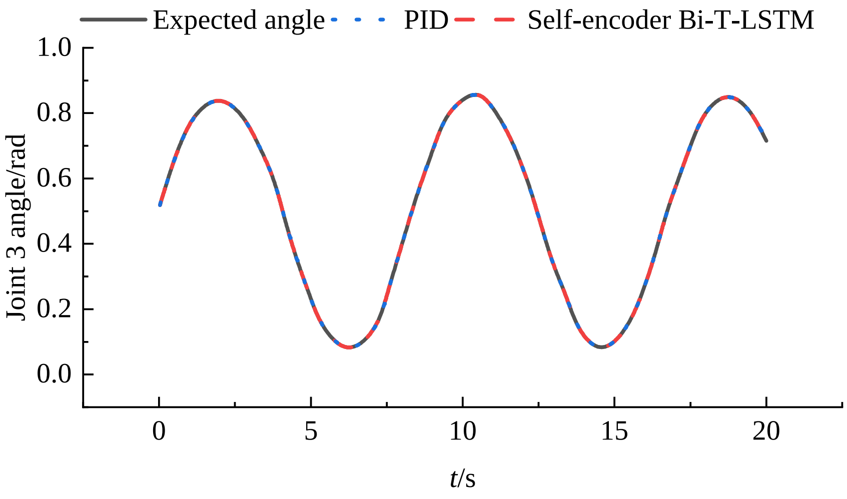
$$l_3 = -965.329 + 301.068t^2 + 9.401t^3, \quad t \in [0, 5] \tag{35}$$

$$\theta_4 = 76.223 - 5.061t^2 + 3.509t^3, \quad t \in [0, 5] \tag{36}$$

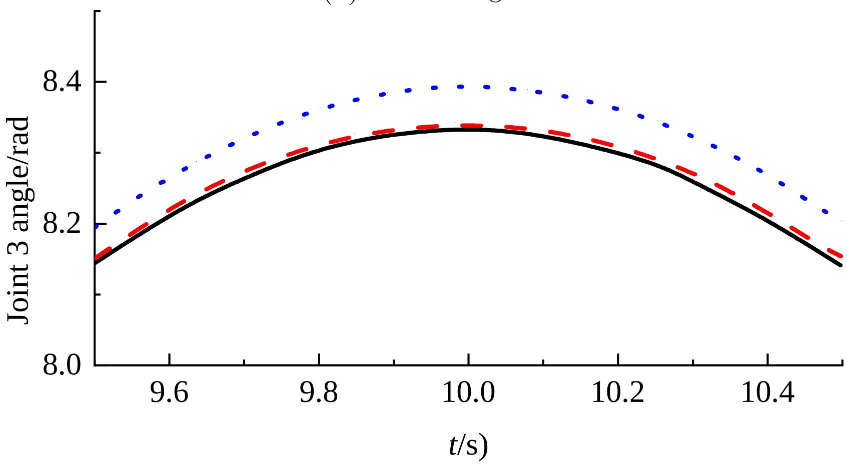
With the proposed controller, the contradiction between the smoothness of motion and the rate of motion during real-time trajectory planning of the sampling robotic arm can be well solved. Taking joint 3 as an example, Figure 6 depicts the comparison of sampling robotic arm joint tracking under neural network control and PID control.

It can be seen that the proposed control algorithm ensures that the joint angles satisfy the system constraints and that the tracking error converges to a small zero domain. The proposed Self-encoder Bi-T-LSTM neural network controller has better tracking control than the PID controller. By sampling the positional tracking comparison of the robotic arm in the task space, as with the PID controller, it can be found that the Self-encoder Bi-T-LSTM neural network controller can also achieve accurate positional tracking control.

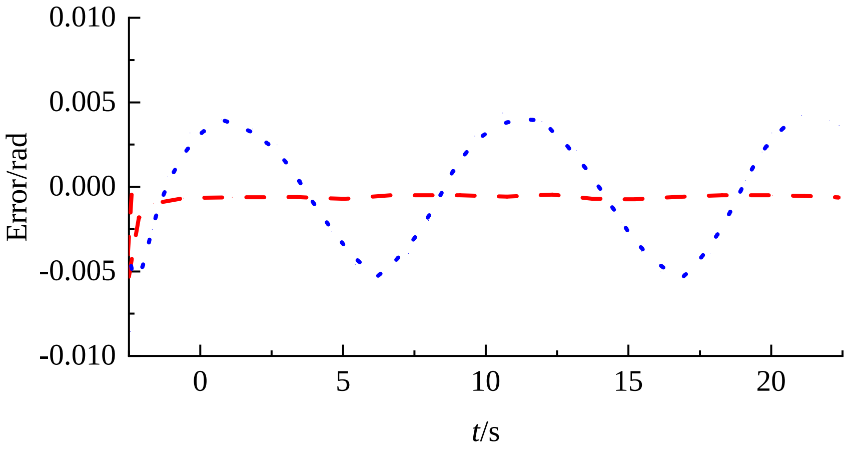
5. **Conclusion.** In order to solve the problem of difficult control when the sampling robotic arm is working, the implicated kinematic relationship between each linkage of the sampling robotic arm is studied. The linkage coordinate system of the neighbouring rods is established for the sampling robot arm, and the kinematic equations of the sampling robot arm are further established by using D-H transformation. In order to further explore the intrinsic connection between the current record and the records of the past and future moments, a Self-encoder Bi-T-LSTM is proposed, which can deepen the level of feature extraction for the original sequence, and thus improve the precision of the trajectory control of the robotic arm. A robotic arm controller based on Self-encoder Bi-T-LSTM is designed. A neural network controller is used to virtually control



(a) Joint 3 angle



(b) Local amplification



(c) Angle error

Figure 6. Joint tracking of Self-encoder Bi-T-LSTM and PID control

the input α_2 to compensate for the unknown mechanical system dynamics, and the corresponding neural network weight control law is designed. The simulation results show that Self-encoder Bi-T-LSTM has outstanding performance in handling time-series data with irregular time interval characteristics, and is good at capturing time-dependent and dynamically changing robotic arm modelling tasks. The designed controller can achieve good position tracking control. The main caveat is that the learning speed of Self-encoder Bi-T-LSTM is relatively slow because it takes into account both forward and backward contextual information. However, the task of real-time control of a robotic arm has certain speed requirements. Subsequent attempts will be made to carry out improvements of the simplified version of Bi-LSTM, such as using fewer hidden units or convolutional structures to speed up.

Acknowledge. This work was supported by Guidance Project of Hubei Provincial Department of Education (No. B2020083).

REFERENCES

[1] M. Slamani, A. Nubiola, and I. Bonev, "Assessment of the positioning performance of an industrial robot," *Industrial Robot: An International Journal*, vol. 39, no. 1, pp. 57-68, 2012.

[2] J. Arents, and M. Greitans, "Smart industrial robot control trends, challenges and opportunities within manufacturing," *Applied Sciences*, vol. 12, no. 2, 937, 2022.

[3] T. Brogårdh, "Present and future robot control development—An industrial perspective," *Annual Reviews in Control*, vol. 31, no. 1, pp. 69-79, 2007.

[4] E. Abele, M. Weigold, and S. Rothenbücher, "Modeling and identification of an industrial robot for machining applications," *CIRP annals*, vol. 56, no. 1, pp. 387-390, 2007.

[5] A. Lopes, and F. Almeida, "A force–impedance controlled industrial robot using an active robotic auxiliary device," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 299-309, 2008.

[6] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87-94, 2012.

[7] M. Švaco, B. Šekoranja, F. Šuligoj, and B. Jerbić, "Calibration of an industrial robot using a stereo vision system," *Procedia Engineering*, vol. 69, pp. 459-463, 2014.

[8] G. Ferretti, G. Magnani, P. Putz, and P. Rocco, "The structured design of an industrial robot controller," *Control Engineering Practice*, vol. 4, no. 2, pp. 239-249, 1996.

[9] D. Massa, M. Callegari, and C. Cristalli, "Manual guidance for industrial robot programming," *Industrial Robot: An International Journal*, vol. 42, no. 5, pp. 457-465, 2015.

[10] J. Gao, H. Zou, F. Zhang, and T. Y. Wu, "An intelligent stage light-based actor identification and positioning system," *International Journal of Information and Computer Security*, vol. 18, no. 1/2, 204, 2022.

[11] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19-46, 2024.

[12] S.-M. Zhang, X. Su, X.-H. Jiang, M.-L. Chen, T.-Y. Wu, "A traffic prediction method of bicycle-sharing based on long and short term memory network," *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17-29, 2019.

[13] Y.-T. Chang, H.-R. Yang, and C.-M. Chen, "Analysis on improving the application of machine learning in product development," *The Journal of Supercomputing*, vol. 78, no. 10, pp. 12435-12460, 2022.

[14] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1-15, 2018.

[15] Z. Majdisova, and V. Skala, "Radial basis function approximations: comparison and applications," *Applied Mathematical Modelling*, vol. 51, pp. 728-743, 2017.

[16] B. Amiri, M. Fathian, and A. Maroosi, "Application of shuffled frog-leaping algorithm on clustering," *The International Journal of Advanced Manufacturing Technology*, vol. 45, pp. 199-209, 2009.

[17] H. Seraji, and B. Bon, "Real-time collision avoidance for position-controlled manipulators," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 670-677, 1999.

[18] L. Ren, J. K. Mills, and D. Sun, "Trajectory tracking control for a 3-DOF planar parallel manipulator using the convex synchronized control method," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 613-623, 2008.

[19] S. Ramabalan, R. Saravanan, and C. Balamurugan, "Multi-objective dynamic optimal trajectory planning of robot manipulators in the presence of obstacles," *The International Journal of Advanced Manufacturing Technology*, vol. 41, pp. 580-594, 2009.

[20] M. Elbanhawi, M. Simic, and R. N. Jazar, "Continuous path smoothing for car-like robots using B-spline curves," *Journal of Intelligent & Robotic Systems*, vol. 80, pp. 23-56, 2015.

[21] Y. Liu, C. Guo, and Y. Weng, "Online time-optimal trajectory planning for robotic manipulators using adaptive elite genetic algorithm with singularity avoidance," *IEEE Access*, vol. 7, pp. 146301-146308, 2019.

[22] G. Giuffrida, G. Meoni, and L. Fanucci, "A YOLOv2 convolutional neural network-based human–machine interface for the control of assistive robotic manipulators," *Applied Sciences*, vol. 9, no. 11, 2243, 2019.

[23] Y. Liu, P. Gao, C. Zheng, L. Tian, and Y. Tian, "A deep reinforcement learning strategy combining expert experience guidance for a fruit-picking manipulator," *Electronics*, vol. 11, no. 3, pp. 311, 2022.

[24] Y. Shi, W. Zhao, S. Li, B. Li, and X. Sun, "Direct derivation scheme of DT-RNN algorithm for discrete time-variant matrix pseudo-inversion with application to robotic manipulator," *Applied Soft Computing*, vol. 133, 109861, 2023.

[25] P. Suebsombut, A. Sekhari, P. Sureephong, A. Belhi, and A. Bouras, "Field data forecasting using LSTM and Bi-LSTM approaches," *Applied Sciences*, vol. 11, no. 24, 11820, 2021.

[26] J. Wang, B. Tao, Z. Gong, S. Yu, and Z. Yin, "A mobile robotic measurement system for large-scale complex components based on optical scanning and visual tracking," *Robotics and Computer-Integrated Manufacturing*, vol. 67, 102010, 2021.

[27] H. Wang, X. Wang, W. Yang, and Z. Du, "Design and kinematic modeling of a notch continuum manipulator for laryngeal surgery," *International Journal of Control, Automation and Systems*, vol. 18, pp. 2966-2973, 2020.

[28] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia CIRP*, vol. 99, pp. 650-655, 2021.