The International Journal of Advanced Manufacturing Technology (2024) 135:5253–5265
https://doi.org/10.1007/s00170-024-14794-z

ORIGINAL ARTICLE

Neural network–based transfer learning to improve stiffness modeling of industrial robots with small experimental data sets

Kai Wu[1] · Yuanhui Zhang[1] · Dehua Gao[1] · Shuhan Deng[1] · Weihua Li[1] · Mingfeng Wang[2]

Received: 4 September 2024 / Accepted: 4 November 2024 / Published online: 16 November 2024

Abstract

Stiffness modeling is an essential subject for the composition of robot control. Accurate stiffness modeling is helpful for improving the control accuracy of industrial robots, particularly under dynamic load circumstances. The classic virtual joint modeling (VJM) method is challenging in predicting the deformation of the end-effector throughout the full workspace due to the nonlinear deformation of the robot joint and its serial articulated structure. This paper proposes a full-space stiffness modeling method for robots based on the integration of a multi-layer perceptual (MLP) model and VJM. To provide enough training data for the MLP model, VJM is used to build a stiffness model with a small set of experimental data to generate 106,400 training data. A model-based transfer learning approach is proposed to improve the model's accuracy and generalization regarding the difference between generated training data and actual experimental data. The VJM stiffness model is compared with the MLP stiffness model and the existing CNN-based transfer learning model based on the same experimental data. Considering the deformation prediction in the three directions in Cartesian space, the mean absolute error, standard deviation, and maximum error of the MLP model are decreased by at least 24.90%, 14.20%, and 8.50%, respectively, than the VJM. These prediction results demonstrate that the proposed modeling technique can significantly increase the accuracy of robot stiffness modeling, which is essential for position compensation in precise motion control of robots under dynamic load.

Keywords Artificial neural networks · Industrial robots · Stiffness modeling · Transfer learning · Virtual joint modeling

1 Introduction

Robots are constantly being used in a wide variety of industrial sectors due to their operational flexibility and low economic costs [1, 2]. Industrial robots (IRs) are generally considered to have lower static stiffness compared to machine tools, due to their tandem articulated structures [3, 4]. The stiffness deformation of IRs is an important factor limiting machining efficiency and quality [5, 6]. The stiffness of IRs varies with position, making the stiffness prediction more complex than machine tools. Therefore, constructing an accurate stiffness prediction model to optimize the machining trajectory of IRs is crucial to improving the machining quality and efficiency [7, 8].

✉ Mingfeng Wang
Mingfeng.Wang@brunel.ac.uk

1 Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 511442, China

2 Department of Mechanical and AerospaceEngineering, Brunel University London, London UB3 8PH, UK

The virtual joint modeling (VJM) method is a common theoretical approach for the stiffness modeling of serial industrial robots, which has the advantages of simplicity and low computational effort [9, 10]. Many researchers have contributed to the application and optimization of the VJM. Dumas et al. contributed a new method to identify the joint stiffness of industrial robots. The external torque was applied to each joint separately, where the rotational deformation was measured, and the joint stiffness was calculated. Meanwhile, the influence of the VJM's supplementary matrix on the overall matrix was analyzed and displayed. The results showed that this method can approach the joint stiffness value well with a certain robustness [11]. Schneider et al. analyzed the joint torque of joint deformation and found that the joint torque affected the stiffness of each joint. The method of multi-segment curve fitting was proposed to optimize the fixed joint stiffness in the VJM, which improves the fitting accuracy of the model to a certain extent [12, 13]. As more factors are considered in the VJM, the accuracy is limited by the difficulty of modeling factors and the accuracy of parameter identification.

Artificial neural networks (ANNs) are capable of learning complex mapping relationships between inputs and outputs, which have been shown to be well adapted to nonlinear problems [14, 15]. Through the network structure and parameter tuning of ANNs, relatively favorable results are achieved in the field of [16, 17]. In robotics research, ANNs have been applied in path planning [18, 19], joint stiffness identification [20], and robot eigenfrequency prediction [21, 22]. To get effective model predictions, neural network models must be trained on a substantial amount of data. However, for industrial robots, whose stiffness varies with the dynamic characteristics of the position, there is a great challenge in acquiring a large amount of training data.

As a new machine learning framework, transfer learning (TL) can be applied to solve less labeled target domain problems by transferring existing source domain knowledge. TL provides an effective way to address the reliance on large amounts of experimental data in neural network models [23]. From the perspective of the mechanisms used to bridge the generalization error between the target and source domains using TL techniques, the TL approaches are classified into three groups: instance-based TL, model-based TL, and feature-based TL [24]. The model-based TL approach takes the neural network model as an object for cross-domain transfer and is suitable for scenarios where the source domain data is sufficient, and the target domain labels are scarce [25]. The multi-layer perceptron (MLP) as a typical deep neural network with simple structure and easy training has achieved favorable performance in classification and regression issues, which is particularly suitable for regression fitting of serial data and structured data [26]. The convolutional neural networks (CNN) and its deformed residual network (Resnet) network also have been applied by researchers to numerical regression tasks like remaining useful life prediction [27, 28]. The variability of these network structures makes a difference in performance, such as convergence and precision in fitting regression tasks. Li et al. used the transfer learning approach based on CNN and MLP to achieve fault diagnosis in target domains with different levels of variability, the results showed that the model with transfer learning achieved better results than the model without no-transfer learning, and the CNN-based model transfer learning is more effective compared to the MLP model [29]. Fu et al. achieved cycle capacity estimation of different types of batteries under various working conditions using an MLP-based transfer learning approach [30]. Wu et al. combined transfer learning with deep learning models (including MLP, recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU)) to predict stock prices, and the results showed that MLP has better prediction accuracy [31]. Ye et al. designed an adaptive domain adversarial neural network with dual regressions (ADANN-2R) for robot deformation prediction. The experimental results show that the proposed ADANN-2R model can obtain higher prediction accuracy with less actual data than the traditional stiffness model [32]. Maqsood et al. used TL to classify images of patients with Alzheimer's disease by fine-tuning the pre-trained convolutional network AlexNet. The algorithm gives the best overall accuracy of 92.85% for multi-class classification of unsegmented images [33]. In the field of intelligent fault diagnosis, Zhao et al. established a standard open-source framework based on deep transfer learning (DTL), which has been applied by other researchers for regression fitting task training [34].

Based on the above literature review, the common theoretical method of VJM has significant complexity in parameter identification, and ANNs rely on extensive data in the robot stiffness modeling process, which is impractical for accurate and efficient robot stiffness modeling. Besides, it is challenging to realize stiffness prediction using small sets of position data due to the nature of robot stiffness variation with position. In this paper, a sequential training approach in model-based TL is used to solve the stiffness fitting prediction issue for a small set of data of the target domain. The VJM stiffness model of the robot is constructed by a small set of experimental data. A significant set of simulation data as the source domain data for the transfer learning process is generated by the VJM stiffness model and used for pre-training of the neural network models. The partial experimental positional posture data are used to fine-tune the pre-trained models. The stiffness deformation is also predicted using the deep transfer learning method based on CNN networks established in [34]. Moreover, the prediction results based on VJM, MLP, and CNN are analyzed for errors. The accuracy and efficiency of the robot stiffness model have been significantly improved, and the dependence on experimental data has been reduced by using a transfer learning approach to modeling. The effectiveness of the transfer learning approach to robot stiffness modeling is demonstrated, and an efficient and accurate approach for robot stiffness modeling is presented, which is a key part of perfecting the theory of industrial robots and improving their practical processing capabilities.

The main theoretical and practical contributions made in this paper to the study of the robot stiffness modeling issue are as follows:

1) A deformation prediction approach for the end-effector of robots using deep learning networks is proposed. The robotic VJM is established using end-effector deformation data at different postures with varying loads. Combining the VJM and variable loads in cartesian space and joint space to generate a massive source domain dataset with physical feature meanings that address the dependence of traditional network model training on large volumes of experimental data.

2) A model-based transfer learning approach is proposed to improve the model's accuracy and generalization regarding the difference between generated training data and actual experimental data. Based on the transfer learning approach, the pre-trained model in the source domain is fine-tuned with a small set of experimental data, which improves the model prediction capability at a small cost. A dynamic learning rate adjustment strategy is used in the optimization process of neural network model parameter training.

The contents of this paper are as follows: Sect. 2 mainly introduces the modeling methodology and related theories. Section 3 plans the experiments and generates robot stiffness modeling data, while Sect. 4 establishes an MLP neural network stiffness model using the simulation data. In Sect. 5, the fine-tuning approach is applied to optimize the pre-training MLP model, and the prediction accuracy of different stiffness models is compared and analyzed. The last section summarizes the research.

2 Stiffness modeling methodology

The proposed modeling methodology is illustrated in Fig. 1. To build the stiffness model, experiments are first planned to obtain the robot deformation data under different loads. Then, the experimental data is processed to establish the VJM stiffness model, which is used to generate enormous simulation data for the robot deformation under different robot poses and external loads. A common MLP network is designed to build the pre-training model with this simulation data. Although there are errors between the simulation data and the actual experimental data, the structure of the MLP model can be designed in good coherence to fit the nonlinear distribution characteristics of the stiffness in the entire workspace. Thus, the fine-tuning approach is adopted to adjust the parameters of the pre-trained MLP stiffness model based on actual experimental data to improve its prediction accuracy. The deep transfer learning method based on the framework of the CNN networks was also used to compare the reliability of the results of this approach.

2.1 MLP and CNN model structure

The MLP and CNN are used as the base model of transfer learning in this paper. Their fitting prediction and transfer

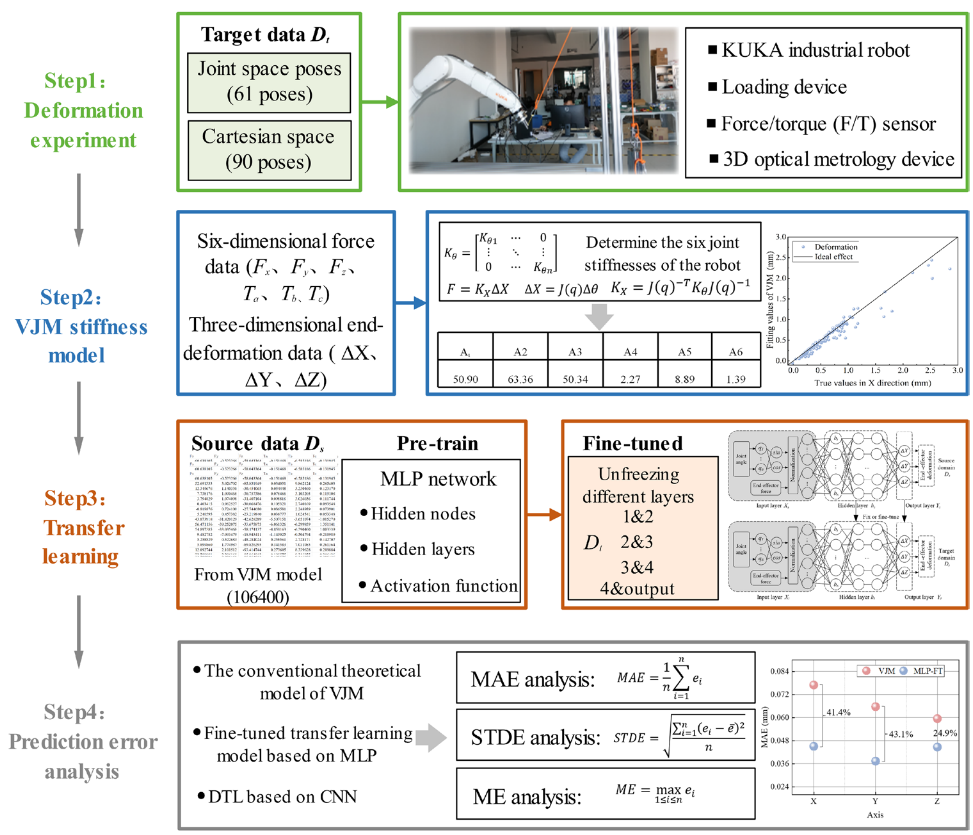

Fig. 1 Illustration of the proposed methodology of robot stiffness modeling using VJM and transfer learning model

learning capabilities are evaluated. The MLP is used as a base model for fine-tuning transfer learning as it achieves relatively favorable results in the numerical fitting. The CNN network, which has four convolutional layers and a multidimensional input and output layer, is used as a base model for deep transfer learning, and its structure is described in detail in [34]. The size of the convolution kernel is one times one. We have modified the output layer of their network so that it can be used to fit multidimensional features. The structures of MLP and CNN neural networks can be seen in Fig. 2, b. We will use the MLP model as a case study to demonstrate the transfer learning process, while the CNN model will only be used as a comparison. The MLP network is formed by arranging multiple neurons in an organized way. The primary artificial neural network model has an input layer, several hidden layers, and an output layer. In this study, the joint angles q_i and the end-effector force F_i, T_i are defined as the input layers, and the three-dimensional deformations (Δx, Δy, Δz) of the end-effector are defined as the output layers. The neurons between different layers use full connectivity. Let x and y be the input vector and output vector of neural networks. The hidden vector is denoted as h_i. Then, the forward propagation process of MLP is as follows:

$$\begin{cases} h_1 = \varphi\left(w_1 x + b_1\right) \\ \quad \cdots \\ h_i = \varphi\left(w_i h_{i-1} + b_i\right) \\ y = func\left(w_{i+1} h_i + b_{i+1}\right) \end{cases} \tag{1}$$

where $\varphi(\cdot)$ is the activation function, w_i is the weight matrix, b_i is the bias vector, and *func* $(\cdot)$ means linear output function. CNN is more complicated than MLP which has convolutional layers, pooling layers, and fully connected layers. The forward propagation process of CNN is as follows:

$$\begin{cases} z_1 = \varphi\left(x * w_1 + b_1\right) \\ c_1 = down\left(z_i\right) \\ \quad \cdots \\ z_i = \varphi\left(c_{i-1} * w_i + b_i\right) \\ c_i = down\left(z_i\right) \\ h_1 = \varphi\left(w_{c1} c_i + b_{c1}\right) \\ y = func\left(w_{c2} h_1 + b_{c2}\right) \end{cases} \tag{2}$$

where $*$ is the convolution operation and down $(\cdot)$ is the pooling operation.

The loss function is a mean squared error (Mse) function for both MLP and CNN, which is as follows.

$$\mathrm{Mse}\left(y_i, \hat{y}_i\right) = \frac{1}{n} \sum_{1=1}^{n} \left(y_i - \hat{y}_i\right)^2 \tag{3}$$

where n is the number of training samples, y_i is one of the truth values, and $\hat{y}_i$ is one of the prediction values. The training method of MLP and CNN is a back propagation (BP) algorithm based on a gradient descent strategy. The purpose of training is to constantly update parameters $\theta = w$, b to minimize the loss function. The updating rule is as follows.

$$\begin{cases} w = w - \alpha \frac{\partial L}{\partial w} \\ b = b - \alpha \frac{\partial L}{\partial b} \end{cases} \tag{4}$$

where α is the learning rate and w and b are the parameters that need to be updated. Learning rate α is a parameter that needs to be set appropriately. A large α may cause a gradient explosion or fail to converge to an optimal value. However, a

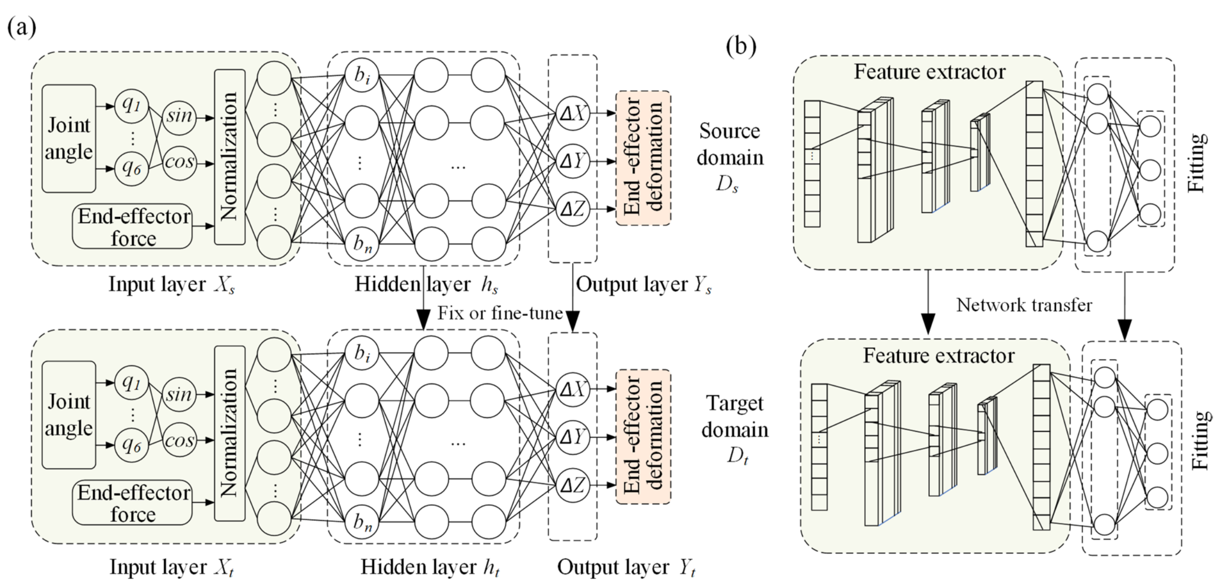

Fig. 2 Structure of MLP network (**a**) and CNN network (**b**) for robot stiffness modeling

small α will make training slow. The essence of the training network is to continuously update the weight w and bias b of the network through the algorithm. In robot stiffness modeling, the network's structure and parameters determine the fitting effect on the training data. Selecting network structures requires hyperparameter tuning and continuous testing for different training sets. The scale and parameter setting of the network are then determined according to the fitting effect. The periodic problem will occur when the angle value is adopted [22], affecting the modeling accuracy. Therefore, the sine and cosine functions are used to process the angles (Eq. 5).

$$\begin{cases} x_i = \sin(q_i) \\ y_i = \cos(q_i) \end{cases} \tag{5}$$

2.2 Transfer learning

In this study, the model-based transfer learning approach was used to realize stiffness prediction modeling. The MLP network was used as the base model to demonstrate the entire methodology flow of pre-training and fine-tuning. Besides, the theory of network-based deep transfer learning as a complementary approach has been explained in detail in Ref. 34, whose open-source network structure is applied to the prediction of target domain data. Stiffness predictions for the robot in different positions are considered as different domains. The MLP model is pre-trained using the source domain data generated by the VJM, and then, the model is fine-tuned with a small set of experimental data. The source domain data $D_s = \left\{\left(x_i^s, y_i^s\right)\right\}_{i=1}^{n_s}$ is the simulation data from the VJM, and the target domain data $D_t = \left\{\left(x_i^t, y_i^t\right)\right\}_{i=1}^{n_t}$ is experimental data. The source domain and target domain consist of n_s and n_t examples, respectively, where $x_i^s, x_i^t \in R^d$ are d-dimensional input vectors, $y_i^s \in R^{d_s}$ is a d_s – dimensional source label vector, and $y_i^t \in R^{d_t}$ is a d_t – dimensional target label vector. The source domains have enough data, but the target domains have only a small amount of data. The task aims to improve stiffness prediction accuracy under different poses. The input vector contains a total of twelve input features, including the six-dimensional force data at the robot's end-effector and the joint angle variable. The output labels of the source and target domains are the three-dimensional deformations of the robot's end-effector. The knowledge of the source domain training task T_s ("pre-train" MLP network) is obtained to help improve the efficiency and effectiveness of the network parameter T_t ("fine-tune" MLP network) of the training target domain.

This transfer learning approach is proposed to optimize the pre-trained MLP network, where "fine-tune" can be called frozen layer fine-tuning. The frozen layer refers to freezing the specified layer parameters of the network, not participating in training, and only modifying the parameters of the unfreezing layer. Fine-tuning refers to setting a lower learning rate and reducing the step size of training. The prior knowledge of the network is reflected in the network structure and parameters. We can regard the learned parameters as an initialization that is close to the optimal solution and then fine-tune them. The difficulty of transfer learning is determining the location of the second training of the network and optimizing the training method.

3 Stiffness modeling data generation

3.1 Experiments design

A six-axis industrial robot (KUKA KR10 R1100-2) is selected for the stiffness modeling evaluation. Its maximum payload is 10 kg, and position repeatability is 0.02 mm. To show the stiffness characteristics of each joint to the greatest extent, the robot poses in this study are divided into two groups: one group is planned in joint space, where the deformation of each joint is recorded after being stressed at different joint angles; the another is the distribution of the robot end-effector in Cartesian space (Table 1).

For joint space poses, the influence of each joint of the robot on the overall posture is different, and the critical joints Axis-2 (A2) and Axis-3 (A3) need to be planned in detail. Based on the measuring range of the 3D optical metrology device, joint angles are planned as follows (Table 1). The Axis-4 (A4) has been set to 0°, and the Axis-5 (A5) and Axis-6 (A6) are adjusted according to the measurement

Table 1 Two kinds of space design schemes

Designed space		A1	A2	A3
Joint space angle (°)		0	− 90 ~ 0	0
			− 120 to − 30	30
			− 150 to − 30	60
			− 150 to − 60	90
			− 180 to − 60	120
			− 180 to − 60	150
		− 15/ − 30	0	0
			− 30	30
			− 30	60
			− 60	90
			− 60	120
			− 60	150
Cartesian space posture	Direction	Range (mm)	Interval (mm)	Number
	X	300 ~ 800	100	6
	Y	− 300 ~ 300	150	5
	Z	300 ~ 800	300	3

requirements. The total number of planned poses in this group is 61.

For cartesian space poses, a cuboid workspace is selected. Measuring positions are planned evenly distributed in the space. There are 90 poses in this group. In joint and Cartesian space, a total of 151 robot poses are planned for the experimental measurement.

3.2 Experimental setup

As shown in Fig. 3, the experimental setup contains a KUKA industrial robot, a loading device, a force/torque (F/T) sensor, and a 3D optical metrology device. The ATI company's axia80-M8 F/T sensor is installed on the robot's flange. Its measurement accuracy can achieve 0.04 N and 0.002 Nm. The C-Track is a 3D optical dynamic tracking system for multiple objects in space. In the volume of 9.1 m^3, its volumetric accuracy is 0.05 mm, and repeatability is 0.01 mm. An aluminum profile and pulley block construct the loading device. Different weights can be added to the hook of the rope.

Each link of the robot is stuck with the round marks, which are tracked by the C-Track. 3D coordinate values referring to the C-Track coordinate system will be recorded when the robot moves to planned positions. All the robot poses are measured.

with the end-effector under unloaded and loaded and the translational deformation can be calculated under different loads.

3.3 VJM stiffness modeling

Based on the experimental data, the corresponding joint stiffness can be calculated. Here, the stiffness of the joint is assumed as a linear spring, and the least square method is adopted to identify the joint stiffness K_θ [11, 12]. Based on the following equation, the Cartesian stiffness K_X can be obtained, where the $J(q)$ is robot Jacobian matrix.

$$K_X = J(q)^{-T} K_\theta J(q)^{-1} \tag{6}$$

The regression diagram is applied to the experimental data and the model fitting data, which is displayed from three directions of XYZ, as shown in Fig. 4. The transverse axis represents the actual value of the deformation, and the longitudinal axis represents the model fitting value of the deformation.

To evaluate the fitting effect, the mean absolute error (MAE), standard deviation of error (STDE), and maximum error (ME) are defined by Eqs. (7)–(9). The fitting results of the three directions are shown in Table 2 according to the above evaluation indexes.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{7}$$

$$\mathrm{STDE} = \sqrt{\frac{\sum_{i=1}^{n}\left(e_i - \bar{e}\right)^2}{n}} \tag{8}$$

$$\mathrm{ME} = \max_{1\le i\le n} e_i \tag{9}$$

where e_i is the absolute value of the error between the model fitting value and the experimental value in a single direction. The results of joint stiffness are shown in Table 3.

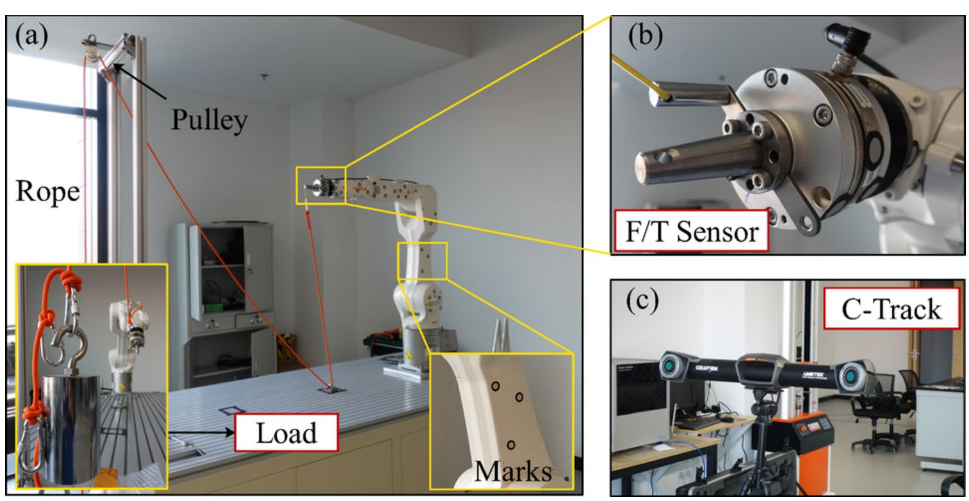

Fig. 3 **a–c** Experiments setup for obtaining translational deformation under specific loads of the robot

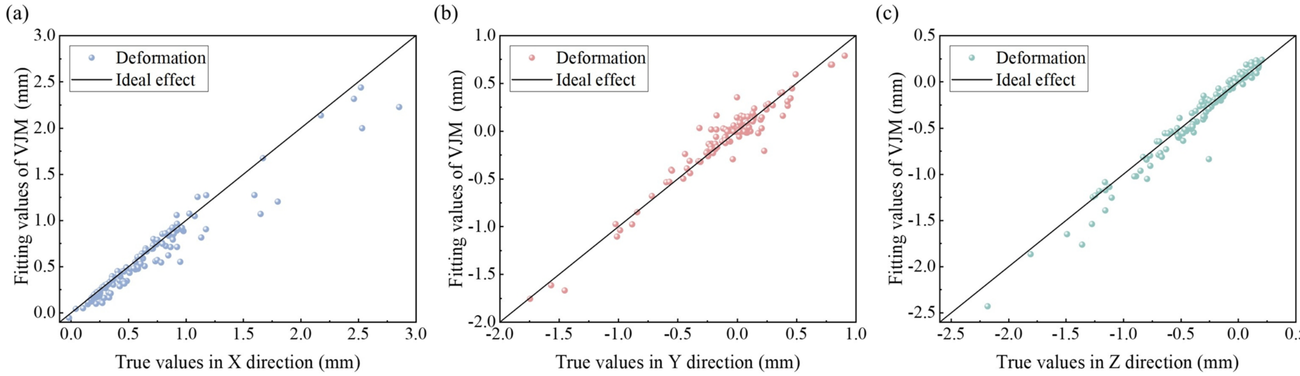

Fig. 4 The VJM fitting effect in different directions

Table 2 Evaluation of fitting results

Direction	MAE (mm)	STDE (mm)	ME (mm)
X	0.074	0.108	0.628
Y	0.066	0.076	0.435
Z	0.062	0.075	0.581

4 Pre-trained MLP stiffness model

4.1 Model input data processing

The input model variables are the six-dimensional joint angle value and the six-dimensional force data, so the input layer contains twelve neurons. The output is the three-dimensional deformation of the robot's end-effector, and the output layer contains three neurons. We use the joint angle space to plan the robot variable pose and apply different loading to generate the source domain dataset.

For joint angle planning, the joint angles have a wide range of motion. Based on the spatial location of the robot's experimental poses, as many robot poses as possible are designed for simulation data generation. The distribution of each joint is shown in the Table 4. Through the pre-modeling simulation analysis, the A4 and A6 are set to 0° to be close to the angle distribution of the experimental data. The planned angles are combined to create a total of 21,280 poses.

For loading force planning, KUKA KR1100-2 is a light-duty robot with a global maximum load of 10 kg. The selected range of the force is designed as shown in Table 5. Five groups of forces are planned to consider the influence of the force in each direction on the end-effector deformation. Then, 106,400 sets of data (including robot poses, loads, and deformations) generated by the VJM simulation are the source domain data for the transfer learning process.

4.2 MLP stiffness modeling

To obtain an optimum model, the main parameters, which are the hidden layers, the layer nodes, and the activation function, are simulated under different settings. This training process sets a variable dynamic learning rate that gradually decreases according to the iteration cycle. The learning rate was set as a dynamic adjustment. Randomly selected 70% of the simulated data was set as the training set and the remaining 30% as the test set. The MLP stiffness modeling and the subsequent transfer learning processes are implemented in PyCharm, and PyTorch is used to build the model training framework. For the number of hidden layers, 2 to 6 layers are selected as the scope of the study. The number of nodes in each layer is set to 100, and the iterative times are 100,000 times. The simulation results indicate that the MLP network with four hidden layers has the best-fitting performance (Table 6).

For the number of nodes, due to the coupling relationship between the number of layers and the number of nodes, 3,

Table 4 The range and indexing value of joint angle

Joint	Begin (°)	End (°)	Interval (°)
A1	−45	45	15
A2	−180	0	10
A3	0	156	10
A5	0	90	10

Table 3 Identification results

Joint	A1	A2	A3	A4	A5	A6
Stiffness (10^3 Nm/rad)	50.90	63.36	50.34	2.27	8.89	1.39

Table 5 The planning of the end-effector force

Groups	Fx (N)	Fy (N)	Fz (N)	Ta (Nm)	Tb (Nm)	Tc (Nm)
1	100	40	– 20	10	10	2
2	80	30	– 40	8	8	1
3	60	20	– 60	6	6	0
4	40	10	– 80	4	4	– 1
5	20	5	– 100	2	2	– 2

Table 6 Fitting results of different hidden layers

Number	Training Set		Testing Set	
	MAE (mm)	STDE (mm)	MAE (mm)	STDE (mm)
2	0.2469	0.1365	0.2501	0.1382
3	0.2164	0.1223	0.2186	0.1230
4	0.2070	0.1178	0.2090	0.1195
5	0.2349	0.1213	0.2375	0.1229
6	0.2381	0.1286	0.2399	0.1318

4, and 5 layers are selected as the basic network structure for the node test. The selection range of nodes is selected from 40 to 400, with an interval of 20. The fitting effect is shown in Fig. 5a, b.

It indicates that the model fitting ability is gradually improved with the increase in the number of nodes, and a structure with 380 nodes is selected for further training. As shown in Fig. 5c, the loss value of the model is constantly decreasing and tends to stabilize with the number of training iterations, which indicates that the model is converging well.

The activation function performs a nonlinear transformation of the model inputs, thus improving the model expression. To evaluate the effect of the activation function, Sigmoid, Tanh, and ReLU functions are selected and trained in the network with four hidden layers and 380 nodes. These activation functions are commonly used in fitting regression problems, but further comparative analysis is needed in this case. Generally, the Tanh function is used when the features of the input data differ significantly, and the Sigmoid function is used when the features of the input data do not vary significantly. The ReLu function enables the output of neurons less than 0 to be zeroed to achieve the effect of a sparse network to fit the function. The fitting results in Table 7 show that the ReLU activation function has better performance than the other two functions under the same training setting. According to the above analysis, further study is based on the MLP networks with settings of 4 hidden layers, 380 nodes, and the ReLU activation function.

5 Transfer learning–based MLP network

To improve the fitting ability and solve the distribution difference between the simulation data and the experimental data, the "fine-tune" transfer learning method is used to optimize the pre-train MLP model.

Table 7 The effect of different activation functions

Activation function	Training set		Testing set	
	MAE (mm)	STDE (mm)	MAE (mm)	STDE (mm)
Sigmoid	0.9656	0.5924	0.9692	0.5936
Tanh	0.3353	0.1959	0.3349	0.1944
ReLU	0.1180	0.0694	0.1199	0.0709

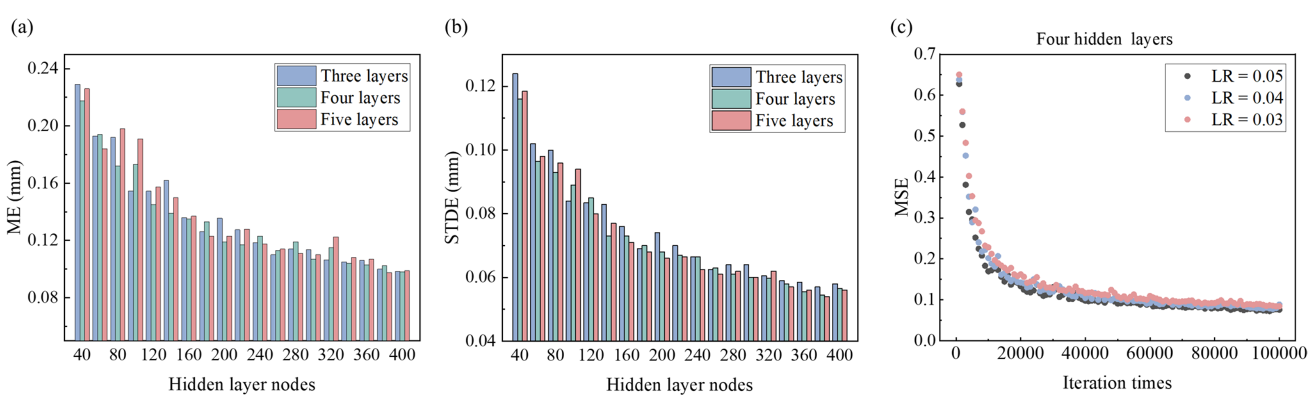

Fig. 5 The influence of hidden layer nodes (**a**, **b**) and the model convergence (**c**)

5.1 Transfer learning of hidden layers

For the MLP network, the number and location of the unfreezing hidden layers must be tested separately from the input and output layers.

To unfreeze the hidden layers, the experimental data were all brought into the network for full training to explore the fitting effect of the network on the data. The strategy of unfreezing the layer is unfreezing one layer only and unfreezing the adjacent two layers. We compare the prediction performance of unfreezing the first and second hidden layers, the second and third hidden layers, the third and third hidden layers, and the output layer of the MLP network. The results of unfreezing one layer were poor compared to unfreezing two layers. Figure 6 shows the regression effects. The results indicate that the training effect after unfreezing two layers is much better than that of one, and it can meet the accuracy requirements. More layers are not apparent for the improvement of the results. Considering all combinations of two layers, the fitting errors of different conditions are presented in Table 8. The results show that the training results of unfreezing the first and second hidden layers are the best. This ufreezing strategy is used to realize subsequent predictions.

Table 8 Unfreezing training results of two layers

Unfreezing Layers	MAE (mm)	STDE (mm)
Hidden layers 1 & hidden layers 2	0.0345	0.0246
Hidden layers 2 & hidden layers 3	0.0564	0.0695
Hidden layers 3 & hidden layers 4	0.0952	0.0326
Output layers	0.1356	0.0772

5.2 Fine-tuned stiffness model

For training method optimization, the structure of the model is determined in the above research. The following discusses how to fine-tune the network parameters appropriately to improve the generalization and prediction ability of the network.

Theoretically, the learning rate (LR) needs to be set low when fine-tuning. However, the lower LR will lead to lower convergence speed. Besides, the network has a good performance fit for the training set, but it does not mean the data prediction ability for the whole set is strong. A coupling of multiple influencing factors exists in this problem. The normal training process will stop after reaching the set convergence standard, adopting a fixed LR and iterations. In this

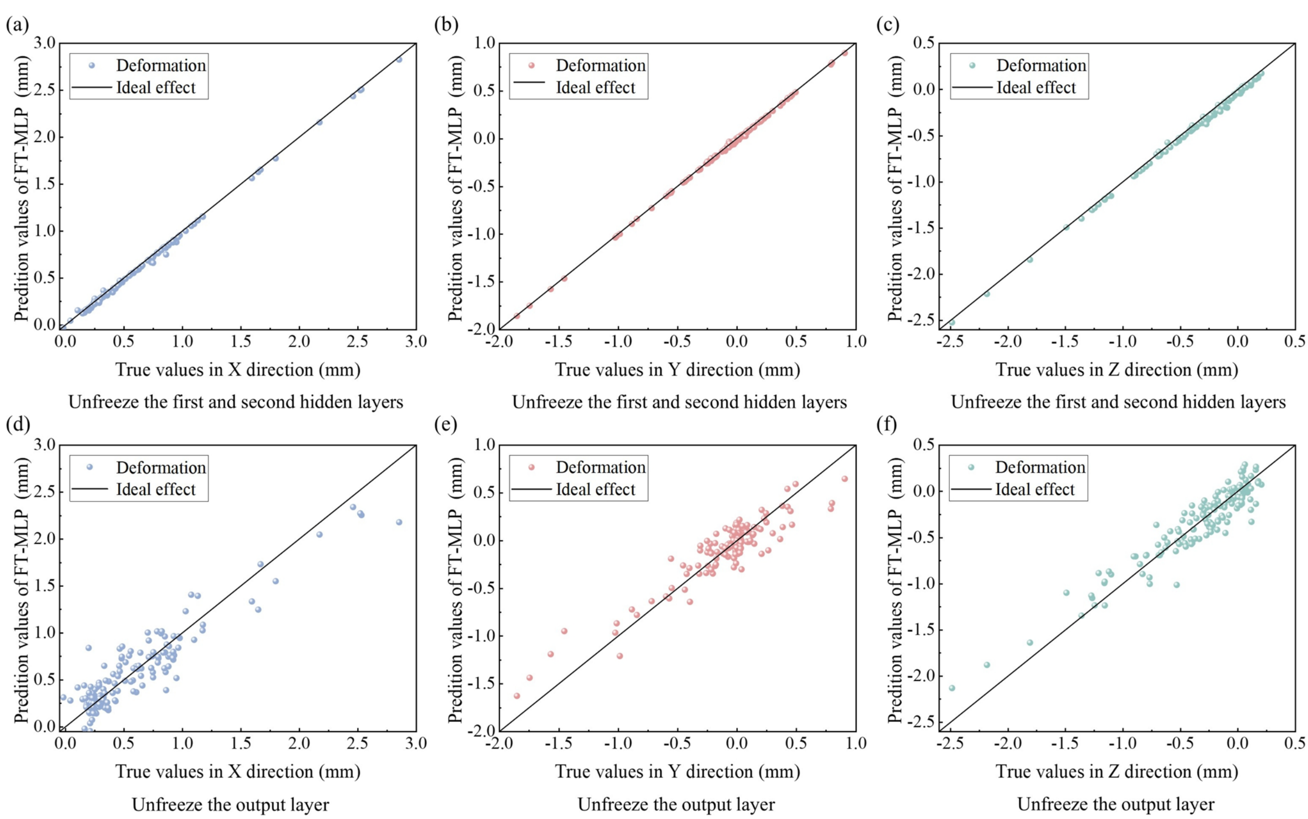

Fig. 6 Comparison of unfreezing different layers in fine-tuned MLP (FT-MLP) process

study, a training strategy is proposed to reduce this effect (Fig. 7). The appropriate network is selected by the fitting effect of the model on the validation set data. Set LR = 0.03, Epochs = 50,000. Twenty percent of the experimental data will be used as a validation set for parameter optimization. Each epoch records and compares the fitting effect of the model validation set, records the best network parameters *M* in the training process, and outputs the optimal network results after completing the training of all epochs.

The modified training method and MAE are used as the loss function to unfreeze the first-layer and second-layer parameters for training. The regression results of the trained model are shown in Table 9. The model's prediction ability is improved based on the proposed training strategy compared to the normal training process. In the testing set, MAE is reduced by 19.70%, and STDE is reduced by 12.00%.

5.3 Different model result comparison

For comparison of modeling results, the VJM stiffness model and a deep transfer learning approach are used to evaluate the performance of the fine-tuned transfer learning stiffness model. In the experiment, 151 robot poses are planned to measure the modeling data. Both models are established by the same data, which is 80% of the experimental data, and the results are evaluated from three aspects, i.e., ME, STDE, and MAE. The prediction results between the three models are shown in Table 10. The prediction results of the deep transfer learning approach based on the CNN network model (DT-CNN) are shown in Fig. 8. The prediction results of the DT-CNN are close to those of VJM, with better prediction results in the *X*-direction. Among them, the prediction results of fine-tuning transfer learning approach based on the MLP model (FT-MLP) are much more favorable than the other two models. The CNN network can extract more advanced and prosperous features through convolutional operations, but the transfer approach does not achieve better results by directly predicting the target domain after training in the source domain. Thus, it is indicated that it is necessary to use the prior knowledge of the source domain and the target domain to fine-tune the pre-trained network model for a better fitting effect. The difference in features between the target and source domains will also have an impact on the fine-tuned effect.

The comparison of the prediction results of FT-MLP and VJM is shown in Fig. 9. It indicates that the accuracy of the FT-MLP is improved in the three aspects compared with the VJM stiffness model. In the *XYZ* directions, the MAE is decreased by 41.40%, 43.10%, and 24.90%, respectively. The STDE decreased by 31.80%, 39.20%, and 14.20%, respectively, and the ME decreased by 8.50%, 21.70%, and 44.20%. The reduction of the STDE and MAE manifest improved prediction generalization ability. The FT-MLP model enhances predictive ability by gaining prior knowledge of

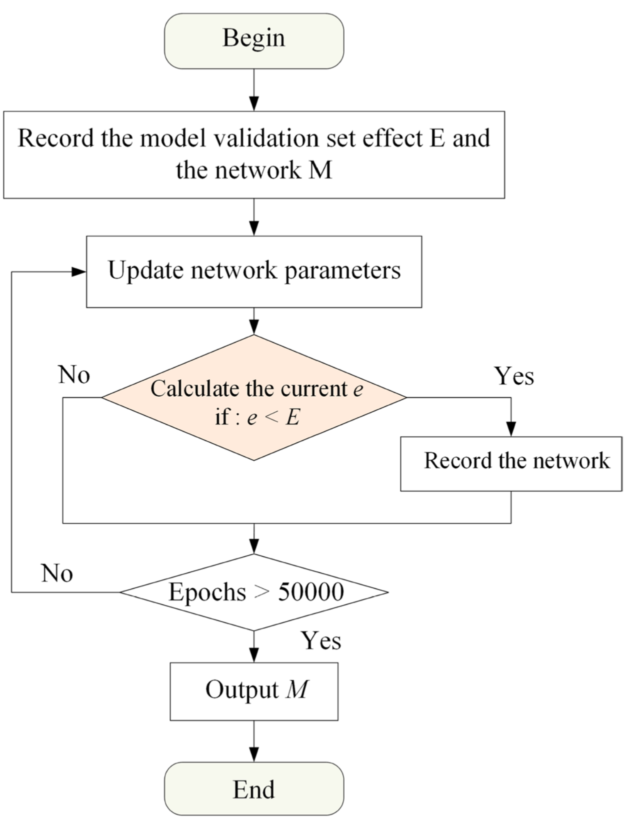

Fig. 7 Training strategy for improving the generalization and prediction ability of the network

Table 9 Comparison of training method

Training method	Training set		Validation set	
	MAE (mm)	STDE (mm)	MAE (mm)	STDE (mm)
Normal	0.0746	0.0584	0.1997	0.1471
Optimized	0.0792	0.0375	0.1604	0.1295

Table 10 Predictive performance of different approaches

Direction	Error (mm)	Approach		
		VJM	FT-MLP	DT-CNN
X	MAE	0.0777	0.0451	0.0729
	STDE	0.1123	0.0766	0.0875
	ME	0.6500	0.6000	0.7700
Y	MAE	0.0657	0.0374	0.0765
	STDE	0.0762	0.0463	0.0928
	ME	0.4300	0.3400	0.5200
Z	MAE	0.0595	0.0447	0.0628
	STDE	0.0717	0.0615	0.0832
	ME	0.5700	0.3200	0.6200

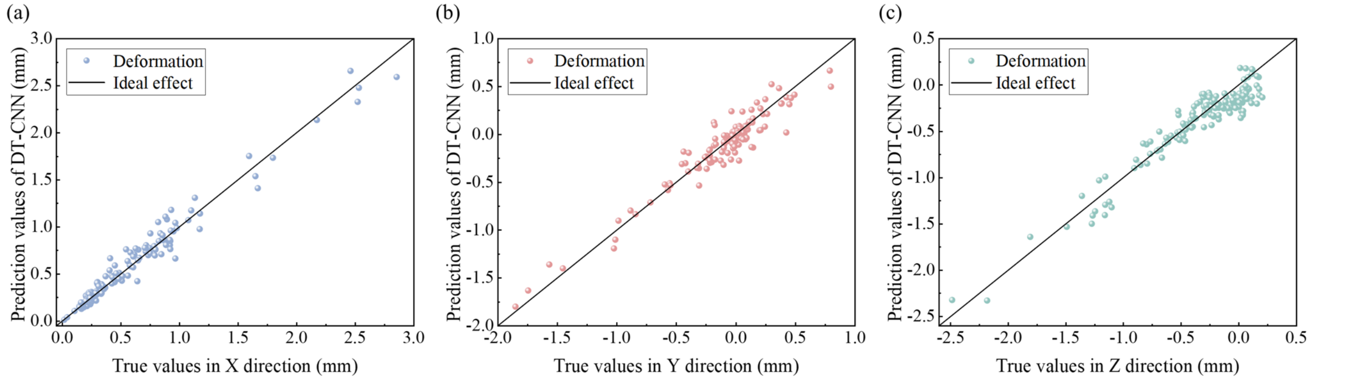

Fig. 8 **a–c** The deep transfer learning prediction results based on the CNN network

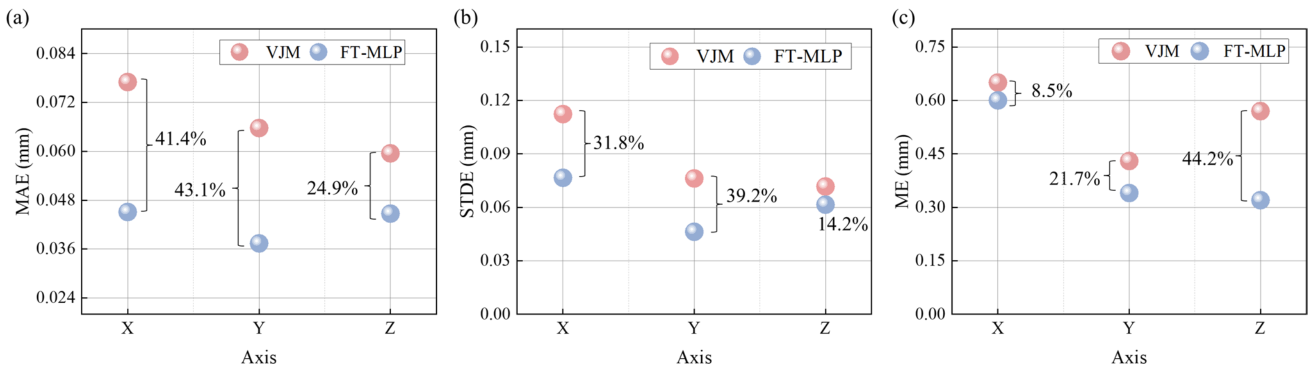

Fig. 9 **a–c** Performance comparison of the FT-MLP model and the VJM

the massive source domain data generated by VJM and by fine-tuning it through experimental data.

6 Conclusion

This paper investigates a model-based fine-tuning transfer learning approach to establish a stiffness model of industrial robots, improving the modeling accuracy and generalization. The VJM is obtained by fitting the experimental data, and a massive set of simulation source domain data is obtained using the VJM. A pre-training model is obtained based on the source domain data, and the pre-training model is fine-tuned using a small set of experimental data.

The predictive ability of the MLP model through fine-tuning is significantly improved compared to the pre-trained model, demonstrating the effectiveness of the transfer fine-tuning strategy. The prediction performance of the proposed approach was compared with the VJM and a deep transfer learning approach based on the CNN network. Results indicate our approach has better performance in improving stiffness modeling accuracy. The mean absolute error, maximum error, and corresponding standard deviation are significantly reduced. The proposed model also provides much better predictive ability than a direct approach using deep transfer learning, demonstrating that utilizing a priori knowledge and fine-tuning the model through the target domain is more effective.

The "fine-tune" transfer learning approach is used for parameter adjustment of the pre-trained network. Many factors need to be considered in the parameter adjustment process. Thus, the model needs to be optimized further for real time and convenience. The subsequent research would exploit an automatic learning approach to optimize the modeling process and improve the model performance. Besides, the next step in the research could be to apply the approach to robot path planning to verify its effectiveness in improving robot machining accuracy. Moreover, the generalization performance of the proposed approach between different robots needs to be validated and refined in future research.

The proposed approach is essential for position compensation in precise motion control of robots under dynamic load. For example, dynamic compensation for robot end-effector deformation is valuable for position control accuracy in high-load tasks such as machining and picking. However, the effectiveness of the proposed approach relies on accurate online sensing of force signals, which deserves more intensive investigation in the future.

Funding The study was funded by the Ministry of Education of the People's Republic of China (CN) (HZKY20220104) and the Brunel University London (12495103).

Declarations

Competing interests The authors declare no competing interests.

References

1. Liu Z, Liu Q, Xu W, Wang L, Zhou Z (2022) Robot learning towards smart robotic manufacturing: a review. Robot Cim-Int Manuf 77:102360
2. Zhang S, Huang H, Huang D, Yao L, Wei J, Fan Q (2022) Subtask-learning based for robot self-assembly in flexible collaborative assembly in manufacturing. Int J Adv Manuf Technol 120(9–10):6807–6819
3. Wan S, Li X, Su W, Yuan J, Hong J (2020) Active chatter suppression for milling process with sliding mode control and electromagnetic actuator. Mech Syst Signal Pr 136:106528
4. Gao H, Shen H, Yu L, Wang Y, Li R, Nazir B (2021) Milling chatter detection system based on multi-sensor signal fusion. IEEE Sens J 21(22):25243–25251
5. Zhu Z, Tang X, Chen C, Peng F, Yan R, Zhou L, Li Z, Wu J (2020) High precision and efficiency robotic milling of complex parts: challenges, approaches and trends. Chin J Aeronaut 35(2):22–46
6. Song Y, Liu M, Lian B, Qi Y, Wang Y, Wu J, Li Q (2022) Industrial serial robot calibration considering geometric and deformation errors. Robot Cim-Int Manuf 76:102328
7. Wu K, Li J, Zhao H, Zhong Y (2022) Review of industrial robot stiffness identification and modelling. Appl Sci-Basel 12(17):8719
8. Liao Z, Wang Q, Xie H, Li J, Zhou X, Pan T (2022) Optimization of robot posture and workpiece setup in robotic milling with stiffness threshold. IEEE-Asme T Mech 27(1):582–593
9. Zhu Z, Chen C, Peng F, Duan X, Wei D (2021) Identification of joint position-dependent stiffness parameters and analysis of robot milling deformation. I Int J Adv Manuf Technol 118(11–12):4179–4193
10. Hu M, Sun H, Liao L, He J (2022) FESM-based approach for stiffness modeling, identification and updating of collaborative robots. Ind Robot 50(1):35–44
11. Dumas C, Caro S, Garnier S, Furet B (2011) Joint stiffness identification of six-revolute industrial serial robots. Robot Comput-Integr Manuf 27(4):881–888
12. Schneider U, Momeni-K M, Ansaloni M, Verl A (2014) Stiffness modeling of industrial robots for deformation compensation in machining. IEEE International Conference on Intelligent Robots and Systems, pp 4464–4469. https://doi.org/10.1109/IROS.2014.6943194
13. Schneider U, Posada JRD, Verl A (2015) Automatic pose optimization for robotic processes. IEEE Int Conf Robot Autom 2015:2054–2059
14. Biswas R, Lourderaj U, Sathyamurthy N (2023) Artificial neural networks and their utility in fitting potential energy curves and surfaces and related problems. J Chem Sci 135(2):22
15. Liu E, Yu Z, Song Y, Sun K, Huang H, Yin F, Zhou Y, Xu K (2023) TL-ANN based nonlinear equalization for multi-users radio over fiber system. J Lightwave Technol 41(5):1399–1405
16. Nikoo M, Malekabadi R, Hafeez G (2023) Estimating the mechanical properties of heat-treated woods using optimization algorithms-based ANN. Measurement 207:112354
17. Mustafa H, Mustapha A, Hayder G, Salisu A (2021) Applications of IoT and artificial intelligence in water quality monitoring and prediction: a review. Proceedings of the 6th International Conference on Inventive Computation Technologies. pp 968–975
18. Mendonca M, Palácios RHC, Breganon R, De Souza LB, Moura LRC (2022) Analysis of the inverse kinematics and trajectory planning applied in a classic collaborative industrial robotic manipulator. IEEE Lat Am T 20(3):363–371
19. Hoai-Nhan HN, Le PN, Kang HJ (2019) A new calibration method for enhancing robot position accuracy by combining a robot model-based identification approach and an artificial neural network-based error compensation technique. Adv Mech Eng 11(1):1687814018822935
20. Almusawi ARJ, Dülger LC, Kapucu S (2016) A new artificial neural network approach in solving inverse kinematics of robotic arm (Denso VP6242). Comp Intell Neurosci 2016: 5720163
21. Das D, Das AK, Pratihar DK, Roy GG (2021) Prediction of residual stress in electron beam welding of stainless steel from process parameters and natural frequency of vibrations using machine-learning algorithms Proc Inst Mech Eng Part C J Mech. Eng Sci 235(11):2008–2021
22. Wu K, Li J (2021) Prediction of the eigenfrequency of industrial robots based on the ANN model. 2021 China Automation Congress (CAC). pp 1595–1598
23. Zhuang F, Qi Z, Duan K, Xi D, Zhu Yong, Zhu H, Xiong H, He Q (2019) A comprehensive survey on transfer learning. Arxiv. pp 27
24. Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big Data 3(1):9
25. Li W, Huang R, Li J, Liao Y, Chen Z, He G, Yan R, Gryllias (2022) A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: theories, applications and challenges. Mech Syst Signal Process 167:108487
26. Prajapati D, Katiyar J, Prakash C (2023) Machine learning approach for the prediction of mixed lubrication parameters for different surface topographies of non-conformal rough contacts. Ind Lubr Tribol 75(9):1022–1030

27. Wang M, Hu W, Jiang Y, Su F, Fang Z (2021) Internal temperature prediction of ternary polymer lithium-ion battery pack based on CNN and virtual thermal sensor technology. Int J Energ Res 45(9):13681–13691
28. Yu W, Pi D, Xie L, Luo Y (2021) Multiscale attentional residual neural network framework for remaining useful life prediction of bearings. Measurement 177:109310
29. Li X, Hu Y, Li M, Zheng J (2019) Fault diagnostics between different type of components: a transfer learning approach. Appl Soft Comput 86:105950
30. Fu S, Tao S, Fan H, He K, Liu X, Tao Y, Zuo J, Zhang X, Wang Y, Sun Y (2023) Data-driven capacity estimation for lithium-ion batteries with feature matching based transfer learning method. Appl Energy 353:121991
31. Wu D, Wang X, Wu S (2021) Jointly modeling transfer learning of industrial chain information and deep learning for stock prediction. Expert Syst Appl 191:116257
32. Ye C, Yang J, Ding H (2022) High-accuracy prediction and compensation of industrial robot stiffness deformation. Int J Mech Sci 233:107638
33. Maqsood M, Nazir F, Khan U, Aadil F, Jamal H, Mehmood I, Song O (2019) Transfer learning assisted classification and detection of Alzheimer’s disease stages using 3D MRI scans. Sensors 19(11):2645
34. Zhao Z, Zhang Q, Yu X, Sun C, Wang S, Yan R, Chen X (2021) Applications of unsupervised deep transfer learning to intelligent fault diagnosis: survey and comparative study. IEEE T Instrum Meas 70:3525828