Proceedings of The 2023 IEEE International
Conference on Real-time Computing and Robotics
July 17-20, 2023, Datong, China

2023 IEEE International Conference on Real-time Computing and Robotics (RCAR) | 979-8-3503-2718-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/RCAR58764.2023.10250091

A Collaborative Robot Torque Prediction Method Based on CNN-TCN Model

Lina Tong[1], Decheng Cui[1], Chen Wang[2*] and Liang Peng[2*]

***Abstract*— The traditional dynamical models show lower accuracy when predicting joint movement, and should be compensated. This paper proposed a model combined with the convolutional network(CNN) and temporal convolutional network(TCN) to compensate for the joint torque prediction values that are calculated from the sensing information. The experiments on the Cooperative Universal Robotic Assistant 6 DoF(CURA6) open dataset, including multi-load and multi-velocity, showed the prediction error can be reduced by 20% compared to other network models. Since there are many kinds of joint movement information, the input data form of the deep learning model should be improved. Thus, the kinetic linearization model is proposed to modify the input of sensing data. According to the different motion types of the CURA6 dataset, comparative experiments were taken, and the mean absolute error was less than 6.8%.**

I. INTRODUCTION

With the development of social intelligence, collaborative robots are widely used in industrial production due to their lightweight structure[1], and it enhances work efficiency. The main working scenario of collaborative robots is to assist workers to complete related tasks[2], so they have strong human-robot interaction. Traditional model-free control methods, such as PID control, have disadvantages such as the inability to accurately track the set trajectory and poor anti-interference ability[3]. Traditional methods cannot meet the requirements of collaborative tasks, such as dragging and teaching[4], [5], collision detection[6], and soft control[7], [8] scenarios. The introduction of robot dynamics systems into control methods to achieve accurate control schemes is becoming more and more widely used. Traditional multi-axis tandem robots for dynamics modelling generally consider the main factors such as gravity, Coriolis force, inertia force and friction[9], but still influence many nonlinear factors, such as gearbox flexibility, motor rotor inertia force[10], etc. This leads to a large error in its calculation of predicted moments, Thus, a robot dynamics system with good performance is a precondition and foundation to ensure its accurate control. Xiao J et al. used a cubic polynomial function fit to the frictional moment term based on traditional dynamics modelling, which resulted in a 25% reduction in the joint torque prediction error[11]. In recent years, deep neural networks have been used in robotics due to their excellent feature learning capability, and Elmar Rueckert et al.[12] used an LSTM network approach to compare with the classical Gaussian process GPs approach to demonstrate the superiority of deep learning methods in feature extraction. wang et al. used a long short-term memory network with an attention mechanism (LSTM) to do compensation for moment errors[13]. However, the long training time of the LSTM network affects the performance. Czubenko M et al. used CURA6 standard robotic arm[14] to collect data sets with multiple loads, multiple speeds and open source and proposed a network model combining convolutional neural network and long short-term memory network, which improves the feature collection capability and has better results for moment prediction but still has the complexity of high shortcomings.

To tackle the above problems, the main target of this paper is to develop combined Convolutional neural networks (CNN) and Temporal convolutional networks (TCN) approach that first exploits the excellent feature extraction capability of CNN networks to obtain more motion information[15]. Secondly, the properties of TCN networks are utilized to capture the temporal dependence[16], since TCNs use causal one-dimensional convolution, dilated convolution and residual layers to increase the size of the perceptual field and solve the problem of computational complexity[17]. The method brings the predicted moments closer to the true values, which are validated using the CURA6 public dataset (http://gitlab.com/intema-gdansk/cura6-dataset), and the cumulative error values in the six axes are reduced by nearly 20% compared to the deep network model proposed by Czubenko M. Moreover, the universality of the proposed CNN-TCN network is demonstrated using motion data of other speeds and loads for validation. Specifically, the main contributions of this work are as follows: (1) the CNN-TCN model is used inside the robotic arm for the first time, (2) Changing the model input is based on dynamics characteristics and performing a rigorous theoretical derivation, (3) Provide a generalized validation scheme for moment prediction of multi-load collaborative robots.

The remainder of this paper is organized as follows: Section II presents the framework of dynamic modelling and moment prediction techniques for robotic platforms. Section III processes the dataset, and Section IV presents the struc-

This work were supported in part by the National Key Research and Development Program of China under Grant 2022YFC3601200 and National Natural Science Foundation of China under Grant U21A20479; and in part by the Beijing Natural Science Foundation under Grants 4232053 and L222013.

[1]School of Mechanical Electronic and Information Engineering, China University of Mining and Technology Beijing, Beijing 100083, China linatong@cumtb.edu.cn, cuidecheng2021@163.com

[2]State Key Laboratory of Management and Control for Complex Systems (SKLMCCS), Institute of Automation, Chinese Academy of Science, Beijing 100090, China wangchen2016@ia.ac.cn, liang.peng@ia.ac.cn

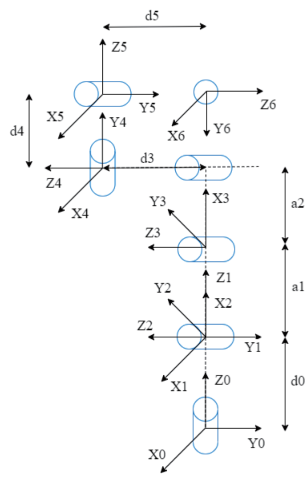
ture and construction method of the deep learning model. Section V evaluates the proposed model for experiments and validates the generalizability, and Section VI summarizes the conclusions.

II. II. CONSTRUCTION OF ROBOT MODEL

A. Dynamic Model of a Robot

The data set used in this study was collected through the general-purpose robotic arm CURA6, which has excellent performance, wide working range and high load capacity, and can achieve acceleration and deceleration motion of 5kg load within 1.2m. The robot platform is shown in Fig. 1, and the modified Denavit-Hartenberg parameter table is shown in Table I. The accurate dynamics model is the basis for realizing functions such as error-free tracking and flexible control. The main methods of robot dynamics representation are Newton Euler method and Lagrange method. According to the consideration of operational efficiency, the Lagrange method is chosen, and its expression is:

(a) Robot Photo



(b) Robot Model

Fig. 1. CURA6 robot model

TABLE I
MDH PARAMETERS

i	α_i	a_i	d_i	θ_i
1	0	0	0.105	θ_1
2	90	0	0	$\theta_2 + 90$
3	0	0.4	0	θ_3
4	0	0.4	0.22	$\theta_4 - 90$
5	-90	0	0.2	θ_5
6	-90	0	0.14	θ_6

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_f \tag{1}$$

$M(q)$ represents the inertia matrix of the robot; $C(q,\dot{q})$ is the matrix of Koch and centrifugal force terms; $G(q)$ is the gravity vector; $q, \dot{q}, \ddot{q}$ are 1×n vectors of robot joint angle, angular velocity, and angular acceleration, respectively; τ_f is the friction force in the robot drive structure, and τ is the robot joint drive torque.

B. Torque prediction method

Among the kinetic parameters, the connecting rod mass, centre-of-mass position, and joint friction are affected by mounting errors and part machining errors. Therefore, direct measurements will have an impact on the final torque prediction results. Compared with the traditional method of constructing the minimum parameter set[18], using deep learning models to accomplish the nonlinear mapping of the dynamics model is one of the key methods to study the dynamics model in recent years. The technical framework of traditional parameter identification methods and deep neural network modelling methods is shown in Fig. 2. Regardless of the method used, the generation of robot motion trajectories is an essential part of the process. A good excitation trajectory can have a wider working space, more velocity variations, and, can make the moment prediction more robust and less variation from the true value. The finite-term Fourier series excitation trajectory proposed by Swevers[19] is widely used. Atkeson[20] and Daemi[21] both proposed to use polynomials as the excitation trajectory using the condition number as the optimization conditions. In this paper, the finite term Fourier series method is used to generate the excitation trajectory to collect the motion information.

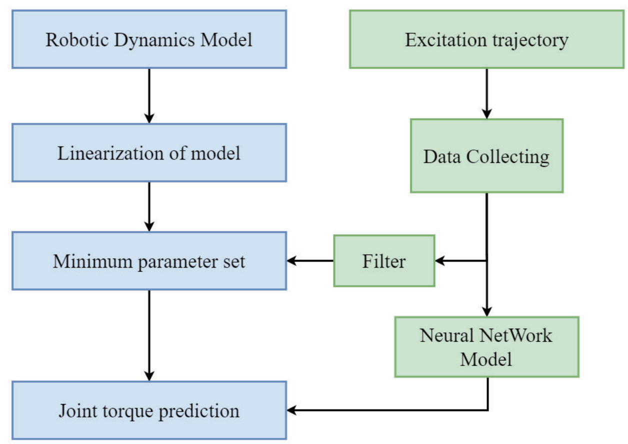


Fig. 2. Torque prediction method process

III. DATA PROCESSING

A. Data classification and serialization processing

CURA6 is an original robot created by Intema in Gdańsk. The dataset used in this paper is collected from the robot motion and contains 91 random motion slices as the training set, each learning slice has about 10,000 samples (acquisition frequency is $24Hz$). There are also 91 shorter random motion clips as the test set. 91 motion clips contain different speeds and different loads of the robot, which can basically cover all the motion situations in the robot usage scenario, and this dataset can be used to evaluate and validate the algorithm with strong persuasion.

In this paper, this dataset is divided into three categories, which are low-speed and multi-load motion set, multi-speed and high-load motion set and multi-speed and low-load motion set, and each set contains 15 motion segments. For the low-speed and multi-load data, the motion speed is 20% of the maximum speed of the joint and below, and the load is from 0kg to 4kg. For the multi-speed and low-load data, the motion speed is 0% to 70% of the maximum speed, and the load is 0.8kg and below. for the multi-speed high-load data, the motion speed is 0% to 70% of the maximum speed, and the load is 3.6kg to 4.1kg. Firstly, the proposed network model is validated by using the low-speed and multi-load motion set, and compared with other network models using this motion set to verify the superiority of the network model in this paper. Secondly, two other motion ensemble datasets are used to verify the generalizability of this network model and demonstrate the robustness of the method on a general-purpose robot.

The TCN model is a prediction model for time series samples, and the data input needs to be serialized, i.e., the moment information of the next moment is predicted by the first n data, and the window sliding process is shown in Fig. 3. This sliding window sample data was collected for the first joint movement of 4 seconds. After the serialization process, the input of the model can be described as batch×windows×feature(number of learning samples×length of windows×number of features).

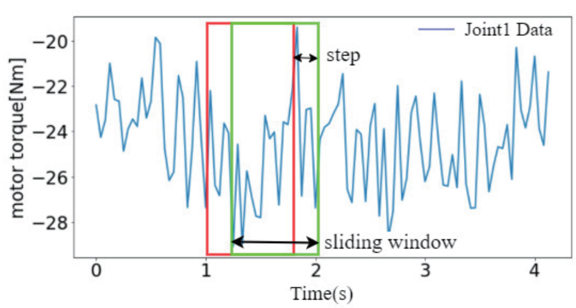


Fig. 3. Sliding window diagram

B. Calculation of joint torque

The input to the network model uses four signals: position of the joint motor, feedback velocity, acceleration and feedback torque, and the output is the predicted torque. The position, velocity and joint torque signals are used without any filtering to preserve the data integrity and real-time. The acceleration signal is low-pass filtered by the feedback velocity at $2Hz$ and then obtained using a central difference algorithm. The relationship between joint motor torque and current can be approximately equal to the linear expression form, as shown in (2). K_1 is the motor torque constant, 0.19 for the first three joints and 0.11 for the other joints, K_2 is the motor gear ratio, 1:100 for this robot, so the final table of joint torque is in the form shown above

$$\tau = K_1 K_2 \cdot I \tag{2}$$

IV. DEEP LEARNING BASED PREDICTION METHODS

A. Convolutional neural networks

Convolutional neural networks(CNN) are mostly used in image processing, video prediction, etc. Sequential signals are considered one-dimensional image information in temporal networks, and CNN networks can also be used to extract more informative features in motion signals according to their network advantages. In the preprocessed data derived in the previous section, the input dimension of batch×16×24 is regarded as a 1×16 one-dimensional image with 24 channels, and feature extraction is performed using a convolutional kernel with a number of 64 and a convolutional kernel size of 1×3. The equation for a one-dimensional convolutional layer is shown[22].

$$Z_i = \sum_{n=1}^{N} b_n^* f_i + b_i \tag{3}$$

where Z_i is the ith output value, b_n is the input feature vector; f_i is the size of the convolution kernel; b_i is the bias on each channel; N is the number of input features; $*$ denotes the convolution operation. The design of multiple convolutional layers and small convolutional kernels can reduce the computational overhead and extract as many shape features at different scales as possible, therefore, increasing the depth of feature information of the data to 64 layers, this behaviour greatly increases the features of motion information and it provides more features for the capture of temporal information by the TCN network later.

B. Temporal convolutional networks

Temporal convolutional networks contain temporal information compared to CNN networks and its prediction of future data extracts features by historical information only. For data with time series, TCN networks add temporal consistency and also have the feature extraction capability of CNN networks. It shows better performance than typical recurrent networks such as Long Short Term Memory (LSTM) in a wide range of datasets and tasks. There are three important modules in TCN networks, which are causal convolution, dilation convolution, and residual linking. The combination of the dilation convolution and residual blocks guarantees the size of the high-feeling field and solves the problem of computational complexity. For a one-dimensional input sequence, it is expressed as

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \tag{4}$$

Dilated convolution operation on the elements in the sequence. In (4), a convolution kernel $f : 0, \ldots, k-1ßR$, where d is the dilation factor, k is the size of the convolution kernel, $s - d \cdot i$ ccounts for the direction of past[23].

In the TCN model design method used in this paper, the size of the convolutional kernel used is 3. To add higher perceptual fields, the expansion factors are chosen as 1, 2, 4, and 8. After passing through the CNN network layers, the input joint motion information is increased to 64 layers, and this information is input to the TCN module with a convolutional kernel size of 3 × 1 × 64. Finally, the neurons are mapped to the 6-axis moment through fully connected layer prediction results. The overall network model diagram is shown in Fig 4. After hyperparametric optimization experiments, the Adam optimizer was set with a training batch size of 512 and a loss function of Mean Square Error(MSE) in the training. 20% was selected as the validation set data and 80% as the training set data. The test set uses the test motion fragment in the dataset. The fairness of the testing effect is ensured. The absolute mean error is chosen as the evaluation index of the model, and the summation of the moments of the six-axis motors is used as the cumulative error for comparison between the models. The Mean Absolute Error(MAE) expression is

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{\tau}_i - \tau_i| \tag{5}$$

Where $\hat{\tau}_i$ represents the actual observed value, τ_i represents the predicted value, and N represents the number of samples.

V. EXPERIMENTAL VERIFICATION

A. Validation of CNN-TCN network model

In order to justify the combination of CNN and TCN networks, the TCN network will be used to compare with the proposed combined network, where the sliding window of different sizes has a decisive effect on the network effect because the motion data of the robot is time-series data. The best results of the Mc-LSTM and CNN-LSTM networks proposed by Czubenko M. are used as the baseline for comparison with this paper, and the sliding window size from 4 to 24. The effect is shown in Fig. 5. It can be seen that the TCN model outperforms the LSTM and its improved networks in processing multi-featured time-series data. The addition of the CNN network before the TCN network is used to extract more feature information. the CNN-TCN network effect is further improved compared with the TCN network. The comparison by sliding windows of different lengths shows that the sum of MAE of six joint moments is the smallest at 20.5 Nm for a window size of 16, and the optimal effect proposed in other papers is 22.5 Nm. it improves the effect by 8.9%.

B. Improvement of input method

In the data pre-processing stage of the network model, the prediction of the network model can be further improved by mining the characteristics of the data itself. The robot dynamics model also has its data characteristics, and (6) is the joint force and moment expression.

$$\begin{aligned}
{}^{i}f_i &= {}_{i+1}{}^{i}R^{i+1}f_{i+1} + m_i{}^{i}\dot{w}_i \times {}^{i}P_{C_i} \\
&+ m_i{}^{i}w_i \times \left({}^{i}w_i + {}^{i}P_{C_i}\right) + m_i{}^{i}\dot{v}_i \\
{}^{i}n_i &= {}_{i+1}{}^{i}R^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}_{i+1}{}^{i}R^{i+1}f_{i+1} \\
&+ {}^{i}P_{C_i} \times m_i{}^{i}\dot{v}_i + {}^{i}w_i \times {}^{i}I_{ii}w_i + {}^{i}I_{ii}\dot{w}_i
\end{aligned} \tag{6}$$

f is the force acting on link $i-1$ and n is the moment acting on link $i-1$. R is the rotation matrix between the joints, P_C is the position of the center of mass of each link, $\dot{v}$ is the linear acceleration of the link, P is the length of the link,w is the angular velocity of the link, $\dot{w}$ is the angular acceleration of the link, m is the mass of the link, and I is the inertia tensor of the link.

It can be seen that the dynamics of joint i are only related to the dynamics and kinematic information of joint $i+1$ and later joints, not to the dynamics and kinematic information of the preceding joints, so the dynamics of the ith joint will be influenced by the dynamics of the later joints and will not be influenced by the dynamics of the preceding joints. (7) – (11) are the matrix expressions of the linearized model.

$$\tau_i = \begin{bmatrix} {}^{i}f_i \\ {}^{i}n_i \end{bmatrix} = {}_{i+1}A \begin{bmatrix} {}^{i+1}f_{i+1} \\ n_{i+1} \end{bmatrix} + k_i \begin{bmatrix} m_i \\ m_i{}^{i}P_{Ci} \\ I_i \end{bmatrix} \tag{7}$$

$${}_{i+1}A = \begin{bmatrix} {}^{i+1}R & 0 \\ {}^{i}P_{i+1} \times {}^{i}_{i+1}R & {}^{i}_{i+1}R \end{bmatrix} \tag{8}$$

$$k_i = \begin{bmatrix} {}^{i}\dot{v}_i & \left[{}^{i}\dot{\omega}_i \times\right] + \left[{}^{i}\omega_i \times\right]^2 & 0 \\ 0 & -\left[{}^{i}\dot{v}_i \times\right] & \left[{}^{i}\dot{\omega}_i\right] + \left[{}^{i}\omega_i \times\right]\left[{}^{i}\omega_i\right] \end{bmatrix} \tag{9}$$

$$\rho_i = \begin{bmatrix} m_i \\ m_i{}^{i}P_{Ci} \\ I_i \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} \tau_1 \\ \vdots \\ \tau_6 \end{bmatrix} = \begin{bmatrix} k_1 & \frac{1}{2}Ak_2 & \cdots & \frac{1}{6}Ak_6 \\ 0 & k_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{5}{6}Ak_6 \\ 0 & \cdots & 0 & k_6 \end{bmatrix} \cdot \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_6 \end{bmatrix} \tag{11}$$

Equation (7) is the form of the kinetic matrix expression for a single joint, and after combining the single joint formulas, the dimension of the moment matrix in (11) is 36×1, the dimension of the observation matrix is 36×60, and the dimension of the matrix composed of the kinetic parameters P is 60×1.

Observation matrix is more intuitively seen as an upper triangular matrix, where the factors influencing the joint dynamics characteristics are only influenced by the right-hand joint information. Therefore, the input to the neural network model is improved, and only the kinematic and moment information of the joint i, joint $i+1$ and later is input to the input of the network model of the joint i. Since the number of model outputs has been changed from 6 to 1 due to the improved input method, the structure of the model can be appropriately simplified. First, the number of convolutional kernel channels is changed to 1/2 of the original number, the size of individual convolutional kernels is changed from 3 to 2, the expansion factors are changed to 1, 2, and 4, and the other hyperparameters are set in the same way. By simplifying the structure, the training parameters of the model are about 14% of the original ones. This approach improves the training efficiency of the

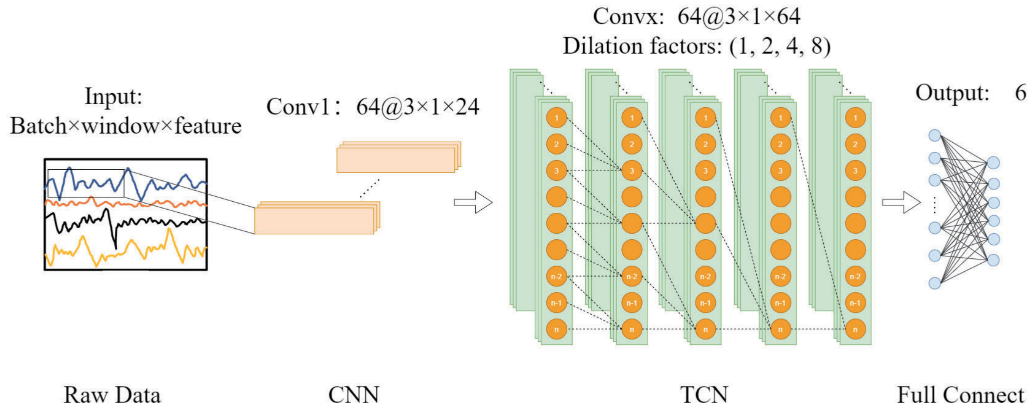


Fig. 4. CNN-TCN prediction model

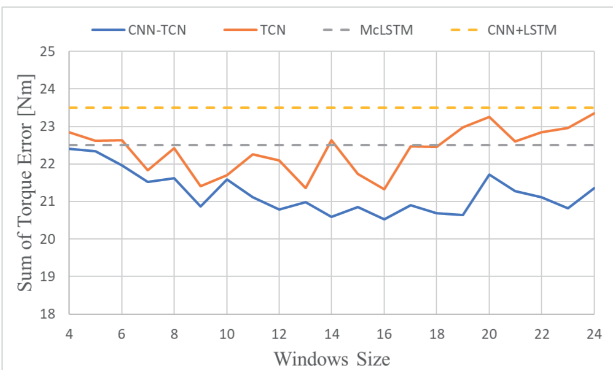


Fig. 5. Model effect comparison chart

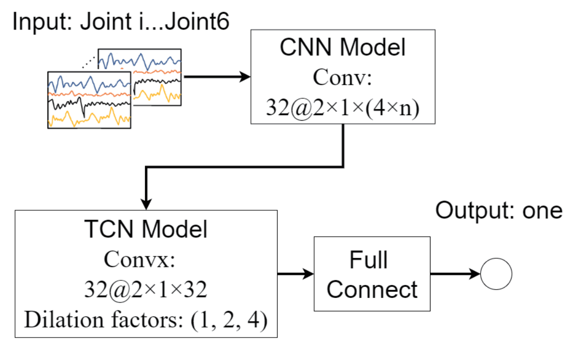


Fig. 6. New network model diagram

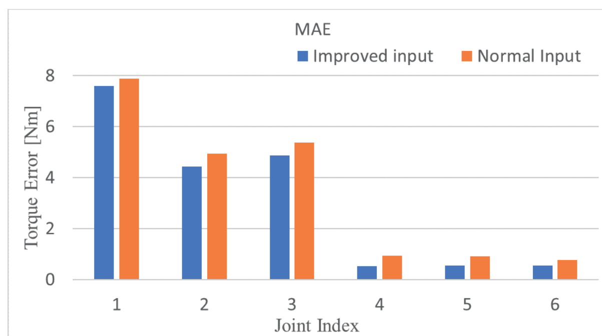


Fig. 7. MAE Comparison Chart

model. Table II compares the changed model structure with the original structure. Fig. 6 shows the framework of the improved network model. After adopting the new input, it is shown in Fig. 7. Joint error of low-speed and multi-load is shown in Table III.

TABLE II
COMPARISON OF THE NEW MODEL STRUCTURE AND THE OLD MODEL STRUCTURE

	Normal Input method	Optimized input method
Input channel	1th, 2th…6th	ith, (i+1)th…6th
number	64	32
Kernel size	3	2
Dilation rate	[1, 2, 4, 8]	[1, 2, 4]
Dropout	0.2	0.2
Activation	Relu	Relu

With the new input method, the sum of MAE is 18.51 Nm, which is smaller than the 20.58Nm of the normal input method. The sum of MAE of torque for 6 joints is reduced by 20% by changing the input method and network model. It can be seen from the table that the error of the main stressed joints is small, less than 5%, the inertia of the end joint is small, and the noise is large, resulting in a large percentage of prediction error, but less than 10%. Finally, the average absolute error of 6 joints is 6.5% of the true torque when using low-speed multi-load data. The error requirement for use was met. Validation of method generalizability.

The above experiments can prove that the CNN-TCN network proposed in this paper outperforms the Mc-LSTM network model proposed by Czubenko M et al.[14] In order to verify the robustness of the method, different data type scenarios are tested in the CURA6 dataset, respectively. The sliding window was chosen to be 16. The joint error of multi-speed and low-load is shown in Table IV and joint error of multi-speed and high-load is shown in Table V.

The table above shows the MAE for each joint and the error as a percentage of the maximum operating torque. The results for the multi-speed low load data show that

TABLE III
LOW-SPEED, MULTI-LOAD VERIFICATION

	1	2	3	4	5	6
MAE	7.58	4.43	4.86	0.53	0.55	0.55
Error	9.4%	4.5%	4.6%	4.6%	6.9%	8.8%

TABLE IV
MULTI-SPEED, LOW-LOAD VERIFICATION

	1	2	3	4	5	6
MAE	8.12	5.68	6.47	0.65	0.577	0.621
Error	7.5%	4.7%	6.3%	6.2%	7.4%	8.6%

the sum of the MAE for the six joints is approximately 22.12Nm, and the average error in joint torque is 6.8%. The results of the multi-speed high load data showed that the sum of the MAE for the six joints was approximately 22.61Nm, and the average error in joint torque was 5.8%. The larger average error in the multi-speed low-load data is due to the larger prediction error caused by the noisy current collection of the joint motor at low load, but this is within a reasonable error range. In summary, the CNN-TCN model shows advantages over other deep neural network models and has good robustness.

VI. CONCLUSIONS

In order to obtain more accurate joint moment prediction results, this paper fully analyzes the motion data features and proposes for the first time a deep network model using CNN combined with TCN for moment prediction. The feasibility of the model is verified using the publicly available dataset CURA6 robotic arm. The results are better than the models of other methods using this dataset. The input method was improved for the unique dynamics of the robotic arm, and a new input method was used, resulting in an overall effect of the model with a 20% reduction in the sum of the moment errors of the 6-axis robotic arm compared to the Mc-LSTM model. To verify the robustness of the model, the motion segments not used in this dataset were divided into a low-speed and high-load motion set and a multi-speed and low-load motion set, and the motion at different loads and different speeds provided a robustness verification scheme. After verification, the results can all achieve less than 6.8% of the maximum error of the torque. In summary, the experimental results in this paper show the effectiveness and robustness of the CNN-TCN model.

TABLE V
MULTI-SPEED, HIGH-LOAD VERIFICATION

	1	2	3	4	5	6
MAE	9.03	5.38	5.21	0.702	0.663	0.63
Error	7.7%	4.7%	4.1%	3.9%	5.5%	8.8%

REFERENCES

[1] A Hentout, M Aouache and A Maoudj. "Human-robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017." Advanced Robotics 33.15-16 (2019): 764-799.

[2] Rozo L, Calinon S and Caldwell D G. "Learning physical collaborative robot behaviors from human demonstrations." IEEE Transactions on Robotics 32.3 (2016): 513-527.

[3] X Shao, S Wang and L Yang. "Research on Feedforward Control Based on Robot Dynamics Parameters Identification." 2020 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2020.

[4] Kushida D, Nakamura M and Goto S. "Flexible motion realized by force-free control: Pull-out work by an articulated robot arm." International Journal of Control, Automation, and Systems 1.4 (2003): 464-473.

[5] Goto, Satoru. "Forcefree control for flexible motion of industrial articulated robot arms." Industrial Robotics: Theory, Modelling and Control; IntechOpen: Mammendorf, Germany (2006): 813-840.

[6] Heo Y J, Kim D and Lee W. "Collision detection for industrial collaborative robots: A deep learning approach." IEEE Robotics and Automation Letters 4.2 (2019): 740-746.

[7] Ott, Christian, Ranjan Mukherjee, and Yoshihiko Nakamura. "Unified impedance and admittance control." 2010 IEEE international conference on robotics and automation. IEEE, 2010.

[8] Yang C, Peng G and Li Y. "Neural networks enhanced adaptive admittance control of optimized robot–environment interaction." IEEE transactions on cybernetics 49.7 (2018): 2568-2579.

[9] Liu G, Iagnemma K and Dubowsky S. "A base force/torque sensor approach to robot manipulator inertial parameter estimation." Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146). Vol. 4. IEEE, 1998.

[10] Li Z, Lun Q and Xing D. "Analysis and implementation of a 3-DOF deflection-type PM motor." IEEE Transactions on Magnetics 51.11 (2015): 1-4.

[11] Xiao J, Zhang Q and Hong Y. "Collision detection algorithm for collaborative robots considering joint friction." International Journal of Advanced Robotic Systems 15.4 (2018): 1729881418788992.

[12] Rueckert E, Nakatenus M and Tosatto S. "Learning inverse dynamics models in o (n) time with lstm networks." 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids). IEEE, 2017.

[13] Wang S, Shao X and Yang L. "Deep learning aided dynamic parameter identification of 6-DOF robot manipulators." IEEE Access 8 (2020): 138102-138116.

[14] Czubenko, Michał, and Zdzisław Kowalczuk. "A Simple Neural Network for Collision Detection of Collaborative Robots." Sensors 21.12 (2021): 4235.

[15] Kattenborn T, Leitloff J and Schiefer F. "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing." ISPRS Journal of Photogrammetry and Remote Sensing 173 (2021): 24-49.

[16] Lara-Benítez P, Carranza-García M and Luna-Romera J M. "Temporal convolutional networks applied to energy-related time series forecasting." applied sciences 10.7 (2020): 2322.

[17] Liu Y, Dong H and Wang X. "Time series prediction based on temporal convolutional network." 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS). IEEE, 2019.

[18] Gautier, Maxime. "Numerical calculation of the base inertial parameters of robots." Journal of robotic systems 8.4 (1991): 485-506.

[19] Swevers J, Ganseman C and Tukel D B. "Optimal robot excitation and identification." IEEE transactions on robotics and automation 13.5 (1997): 730-740.

[20] Atkeson C G, An C H and Hollerbach J M. "Estimation of inertial parameters of manipulator loads and links." The International Journal of Robotics Research 5.3 (1986): 101-119.

[21] Daemi, M., and B. Heimann. "Identification and compensation of gear friction for modeling of robots." ROMANSY 11. Springer, Vienna, 1997. 89-96.

[22] Zang H, Cheng L and Ding T. "Hybrid method for short-term photovoltaic power forecasting based on deep convolutional neural network." IET Generation, Transmission & Distribution 12.20 (2018): 4557-4567.

[23] Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arXiv preprint arXiv:1803.01271 (2018)