IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 4, 2019, accepted November 23, 2019, date of publication December 2, 2019, date of current version December 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957019

Modeling and Simulation of Robot Inverse Dynamics Using LSTM-Based Deep Learning Algorithm for Smart Cities and Factories

NAN LIU[1,2], LIANGYU LI[1], BING HAO[3], LIUSONG YANG[3], TONGHAI HU[3], TAO XUE[2], AND SHOUJUN WANG[2]

[1]School of Mechanical Engineering, Tianjin Polytechnic University, Tianjin 300387, China
[2]National Demonstration Center for Experimental Mechanical and Electrical Engineering Education, Tianjin University of Technology, Tianjin 300384, China
[3]CITIC Heavy Industries Company, Ltd., Luoyang 471003, China

Corresponding author: Shoujun Wang (tjutjun@hotmail.com)

This work was supported by the National Key Research and Development Program of China under Grant 2017YFB1302100.

ABSTRACT In smart cities and factories, robotic applications require high dexterity and security, which requires precise inverse dynamics model. However, the physical modeling methods cannot model the uncertain factors of the manipulator such as flexibility, joint clearance and friction, etc. As an alternative, artificial intelligence (AI) techniques have become increasingly popular in robotics for smart cities and factories. In this paper, deep learning neural network based on LSTM (Long Short-Term Memory) is adopted to predict the manipulator inverse dynamics. This study aims to summarize the influence of the hyper-parameter settings on model performance and to explore the applicability of the LSTM model to joint torque prediction of multiple degrees of freedom series manipulator. Furthermore, the feasibility of using only joint position as input data for torque prediction is verified. Simulation result has shown that, for the proposed deep learning architecture, the effects of the number of maximum epochs on model performance should be prioritized. The effects of the number of hidden nodes on model performance are limited, while prediction accuracy will deteriorate as the number of hidden layers increases. It is proved that it is feasible to predict inverse dynamics when input data is joint position only. The experimental results show that the training time increases with the increase of hidden layers, neurons and epochs.

INDEX TERMS Smart cities and factories, inverse dynamics, robot, green computing, deep learning, LSTM.

I. INTRODUCTION

Smart city is an intelligent city based on internet of things, cloud computing and artificial intelligence technology. It adopts advanced information technology, analyzes the trend of the city, makes quick intelligent responses to urban planning, livelihood policies, social security and other aspects, and realizes the intelligent management of the city. At present, there are many urban problems, such as air pollution, water pollution, garbage pollution, shortage of resources, traffic jam and so on. These problems seriously affect people's life and hinder the development of the city. To solve these problems, it is necessary to build smart cities to improve people's way of life, create a beautiful life and environment, and promote urban development and innovation. Similar to smart city, smart factory is composed of many intelligent manufacturing equipments (including control and information systems), namely several intelligent branches and equipment that is composed of various intelligent components.

Robots used for smart cities and factories have accomplished some easy tasks in structured settings that still require fences between the robots and human to ensure safety. Ideally, robots should be able to work side by side with humans, offering their strength to carry heavy loads while presenting no danger. To achieve this objective, it is necessary to obtain an accurate inverse dynamics model of the robot. Moreover, inverse dynamics has been a valuable piece of information for robotic function such as compliance control, human-robot cooperation, target operation and trajectory planning. But robot dynamics is still difficult to solve some problems, such as manipulator collision avoidance,

The associate editor coordinating the review of this manuscript and approving it for publication was Maod Ma.

light manipulator drag teaching, bionic robot balance. The main reason is that the manipulator inverse dynamics model is not accurate enough. Specifically, a) the dynamic parameters of the robot are inconsistent with the actual physical parameters. b) the centrifugal force and coriolis force are ignored to simplify the model. c) the mechanism of friction is still not clear. d) data acquisition and data filtering errors. With the continuous development of smart city and smart factory related technologies, another way to solve such problems is to use machine learning algorithms to model inverse dynamics [1]. Between 1993 and 2016, there were some researches that used machine learning methods to solve the problem of robot inverse dynamics. Chan proposed a single layer linear network to model the robotic inverse dynamics [2]. Three-layer neural networks are proposed to learn the inverse dynamics for flexible manipulator control [3], [4], [5], [6], [7]. Mori *et al.* proposed a forward-propagation learning rule for a neural network to learn an inverse model of a controlled object [8]. Jiang *et al.* established a dynamic compensator by three-layer neural network for improving accuracy of the dynamic model with identified parameters [9]. In [10], [11], [12], three-layer recurrent neural network is used to perform the inverse dynamics problems. Nguyen-Tuong *et al.* and Cruz *et al.* adopted Gaussian Process Regression method to deal with robot inverse dynamics [13], [14], [15], [16]. Camoriano *et al.* adopted semi-parametric learning method to address this problem [17], [18], [19]. In such researches, models are learned directly from data provided by the system's sensors. Thus, knowledge of the robot's physical properties is not required for the derivation of the inverse dynamics modeling.

Deep learning is a new branch of machine learning, which is already emerging in smart cities and factories. With the rapid development of data science, deep learning methods have been widely used in solving classification and regression problems. For example, traffic prediction problem [20], [21], crude oil price, stock and financial prediction [22], [23], [24], software maintainability metrics prediction [25], skeletal muscle forces prediction [26], process alarm prediction [27], indoor temperature prediction [28], video saliency prediction [29], disease prediction [30], aircraft landing speed prediction [31], air quality prediction [32], building-design energy prediction [33] and wind speed prediction [34]. Compared with the shallow neural network, the deep learning neural network can extract the hidden natural structure and inherent abstract features of data better. Accordingly, the deep learning methods can be the promising methods in modeling manipulator inverse dynamics [35], [36]. Recently, some deep learning methods, such as the DNN (Deep Neural Network) and LSTM (Long Short Term Memory), have been applied for establishing the robot inverse dynamics models. Binyan *et al.* [37] put forward a DNN based manipulator inverse dynamics prediction models, and their case studies verified the proposed model was accurate and stable. Rueckert *et al.* [38] put up with the LSTM based model for manipulator inverse dynamics prediction, and the proposed model could obtain satisfactory prediction performance.

Although deep learning has been also applied to some applications in robot dynamics, there are still many shortcomings. First, the hyper-parameter setting is the key technology of deep learning model. However, investigations of different hyper-parameters among those models and comprehensive comparison studies are rarely reported, especially the manipulator inverse dynamics. Second, the input data used for learning inverse dynamics of manipulator are joint position, velocity and acceleration. The joint position sequences of the joints can be measured directly, and then differentiate them to obtain the velocity and the acceleration. The noise of the collected high-frequency data will cause the abrupt change of the speed and acceleration, which will affect the training effect. Although low-pass filtering can alleviate the above problems, the filtering itself will distort the data to some extent.

Therefore, in this paper, according to the strong ability of deep learning to address the time series problem of robotic inverse dynamics, we found that acceptable prediction accuracy of joint force can be obtained only by using joint position as input training data. Moreover, we proposed a deep learning architecture based on LSTM and compared the effects of different hyper-parameters and different input data dimensions on the prediction accuracy of the learning architecture. For case study, we choose Seven-Degree-of-Freedom Heavy-duty Hydraulic Manipulator to test the simulation performance. In summary, the goals of this study are (1) to summarize the influence of the hyper-parameter settings on LSTM based learning model performance and give some suggestions of hyper-parameter setting for manipulator inverse dynamics; (2) to verify that acceptable torque prediction results can be obtained by using only joint position as input data. In addition, the current method of optimizing the topological structure of the learning model is still mainly using the trial-and-error method, which will lead to excessive time and energy consumption. Moreover, the notion of green computing and wireless sensor network technology [39], [40], [41] has become popular given recent concerns about global climate change and the energy crisis. Therefore, we adopt green computing, which enables the computer to turn off unnecessary function modules when performing repeated network training tasks, to reduce power consumption. The main contribution of this paper is to give some suggestions of hyper-parameter setting for manipulator inverse dynamics and verify that acceptable torque prediction results can be obtained by using only joint position as input data. The precise estimation of robotic joint torque in this paper will play a positive role in its application in smart cities and factories.

The rest of this paper is organized as follows. Section 2 details the LSTM network structure and the proposed manipulator inverse dynamics predictive model based on LSTM. Section 3 describes the simulation evaluation object, training data source and pre-processing, parameter

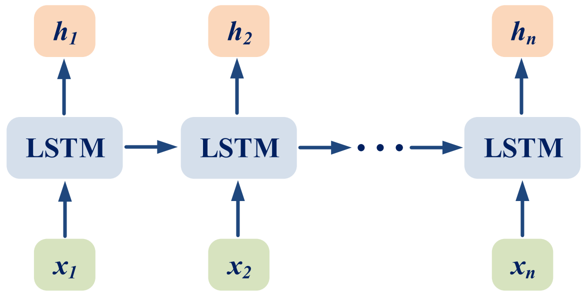


FIGURE 1. Structure diagram for LSTM neural network.

settings of learning model, as well as the evaluation method. Section 4 presents the simulation results, and compares the simulation results from three aspects: prediction accuracy, input data difference and calculation speed. Section 5 concludes the paper with summary and future research prospects.

II. PROPOSED SCHEME

The sequential nature of manipulator inverse dynamics suggests that, to predict the joint torque, it is important to model the inter-relationship among sequential data points. In this section, the LSTM network, an improved variant of the recurrent neural network (RNN), is proposed as the modeling technique for characterizing the inverse dynamics of manipulator.

A. LONG SHORT-TERM MEMORY

In recent years, with the continuous development of deep learning technology, some deep learning models have been gradually applied to the study of time sequent data. Deep learning model is a kind of deep neural network model with multiple nonlinear mapping levels, which can abstract input signals layer by layer and extract features to dig out deeper potential laws. In real life, deep learning is used by search engines to filter content, social media to analyze personal preferences and make recommendations, and various kinds of natural language processing that can be carried on smart devices. The prediction methods used in these successful deep learning applications can be attributed to a branch of artificial neural network, known as LSTM recurrent neural network.

LSTM neural network is one type of recurrent neural network (RNN) with LSTM cells as the hidden layers (As shown in Figure 1). It is worth noting that LSTM cells are the basic unit of LSTM neural network, and the schematic diagram of the unit is shown in Figure 2. The principle of LSTM can be expressed by the following formulas:

$$f_t = \sigma(W_f[h_{t-1}, x_t]^T + b_f) \tag{1}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t]^T + b_i) \tag{2}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c[h_{t-1}, x_t]^T + b_c) \tag{3}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t]^T + b_o) \tag{4}$$

$$h_t = o_t \odot \tanh(C_t) \tag{5}$$

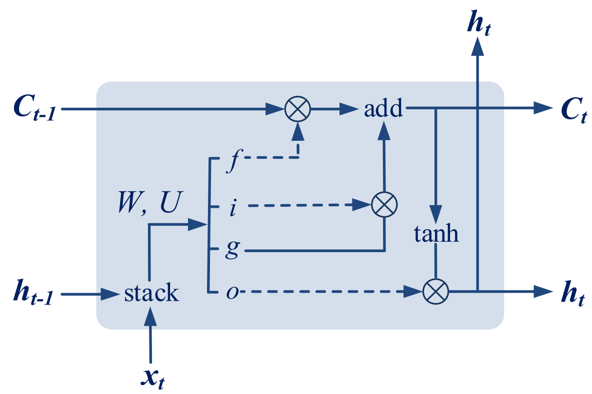


FIGURE 2. Structure diagram for LSTM cell.

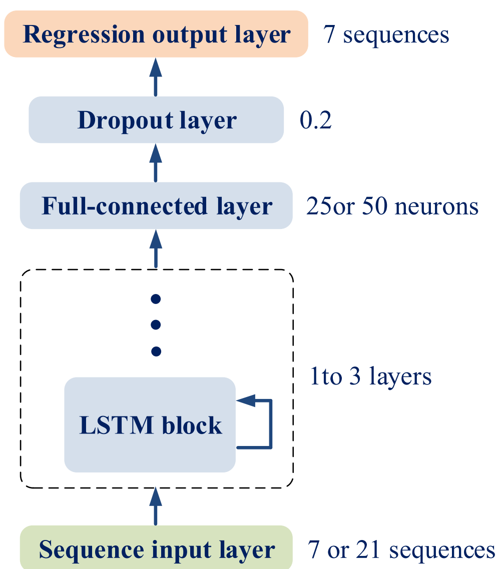


FIGURE 3. Proposed deep learning architecture.

The operator $\odot$ is the dot product of two vectors; $\sigma(\cdot)$ represent sigmoid activation function and $\tanh(\cdot)$ represent hyperbolic tangent activation function. W_f, W_i, W_c and W_o represent the weight matrix of input and gates in the cell, respectively. b_f, b_i, b_c and b_o refer to the bias vectors. h is the output of the cell, and C is introduced as the state of the cell to store the information of the previous cell. f is the factor that determines how much the last state C affects the current cell. x_t is input of current time. h_{t-1} is the output of last cell. At last, h_t of current cell is calculated with C_t and o_t.

B. PROPOSED DEEP LEARNING ARCHITECTURE

In this paper, the proposed architecture of the network has one input layer, one to three LSTM layers, one full-connected layer, one dropout layer and one regression output layer, as shown in Figure 3.

The input layer has 21 neurons (manipulator's 7 joint position, 7 velocity and 7 acceleration, as shown in Figure 4) or 7 neurons (only 7 joint position as shown in Figure 5).

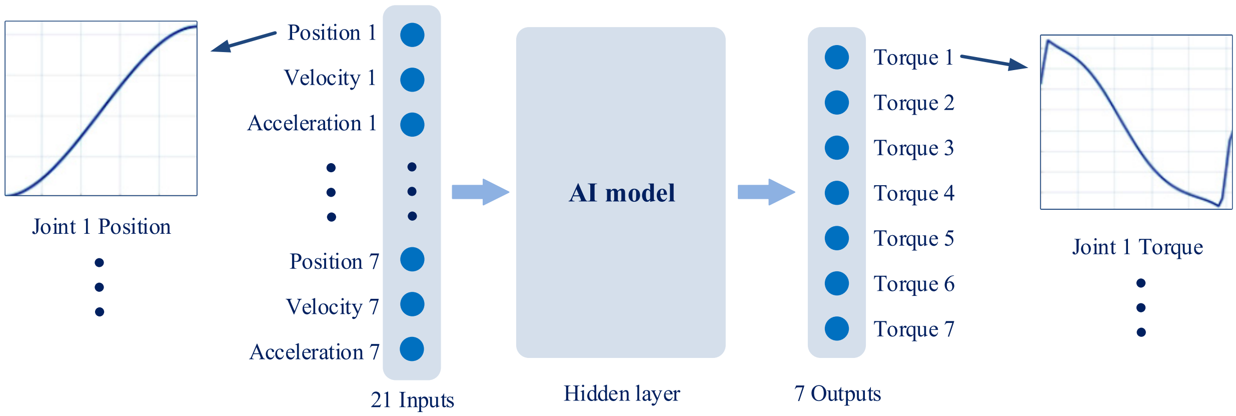


FIGURE 4. **Inverse dynamics learning architecture of manipulator with 21 input data.**

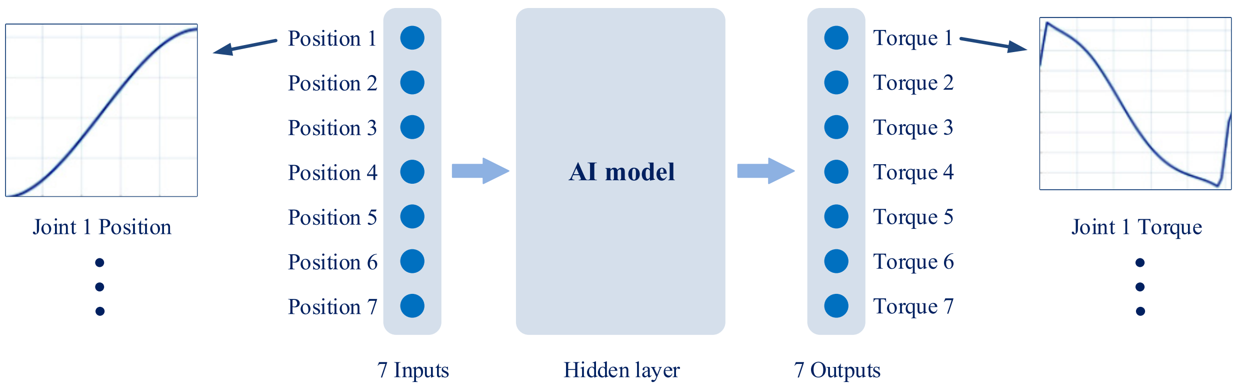


FIGURE 5. **Inverse dynamics learning architecture of manipulator with 7 input data.**

TABLE 1. **Specific settings for the LSTM layers.**

Scheme	Number of layers	Number of neurons	Specific architectural scheme
Scheme1	1	100	LSTM: 100
Scheme2	2	100	LSTM: 66, 34
Scheme3	3	100	LSTM: 58, 28, 14
Scheme4	1	200	LSTM: 200
Scheme5	2	200	LSTM: 133, L67
Scheme6	3	200	LSTM: 114, L57, L29

There are 1 to 3 layers of LSTM, and the total number of LSTM neurons is 100 or 200. The specific settings for the LSTM layers are shown in Table 1. The state activation functions of LSTM cells are set to 'tanh', and the gate activation functions are set to 'sigmoid'. Initialize the input weights with the Glorot initializer. Initialize the forget gate bias with ones and the remaining biases with zeros. The training algorithm adopts back-propagation through time.

Sigmoid activation function:

$$\sigma(x) = \frac{1}{1+e^x} \tag{6}$$

Hyperbolic tangent activation function:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{7}$$

The dropout layer is set to 0.2 to avoid overfitting.

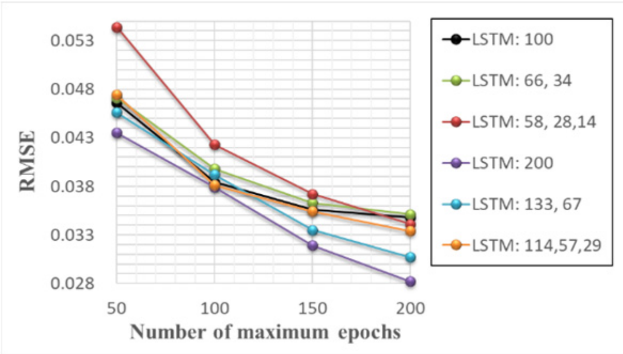
The full-connected layer is directly connected to the output layer and the LSTM layers, which consists of dozens of fully interconnected neurons, specifically, 25 or 50. Initialize the weights with the Glorot initializer and initialize the bias with zeros.

The regression output layer has 7 neurons, which are the torques of 7 joints respectively.

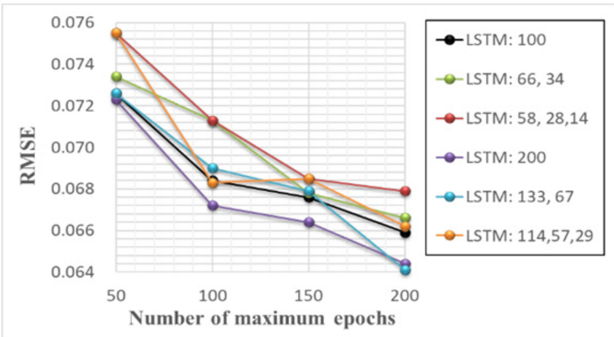
III. PERFORMANCE EVALUATION

A. EXPERIMENTAL SETUP

We evaluated the prediction performance and the computational time for training and generating joint torque predictions in a dynamics model learning task using Seven-Degree-of-Freedom Heavy-duty Hydraulic Manipulator. The input data of AI models are (1) seven joint position, velocity and acceleration, that is 3 parameters for each joint and 21 parameters in total or (2) just 7 joint position, that is only one parameters for each joint and 7 parameters in total. The input data is generated by the step function, which conforms to the start-stop rule of the joint. The output data are 7 joints torque calculated by Newton-Euler method. According to the above rules, we prepared 500 time series as training data, with a total of 173233 points by sampling (at 50 Hz sampling frequency).

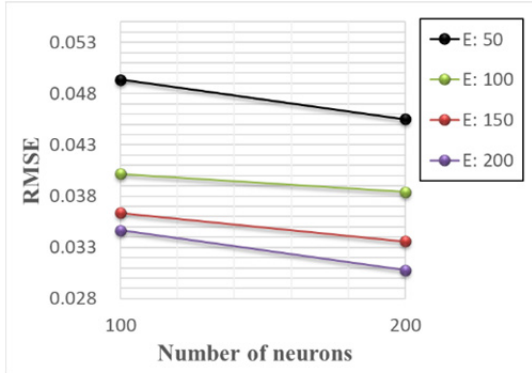


(a) Input: $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$

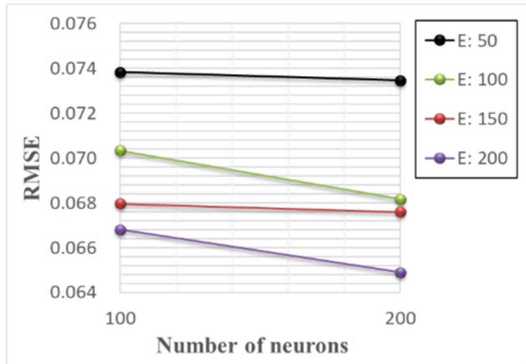


(b) Input: $\boldsymbol{q}$

FIGURE 6. The effect of number of maximum epochs on prediction accuracy.



(a) Input: $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$



(b) Input: $\boldsymbol{q}$

FIGURE 7. The effect of number of maximum epochs on prediction accuracy.

All input and output data are normalized to match the consistency of the learning model. After the prediction of inverse dynamics, the real value is restored. The normalized equation is as follows:

$$x_n = \frac{x_r - x_{min}}{x_{max} - x_{min}} \tag{8}$$

In which, x_n represents the normalized value; x_r denotes the real value; and x_{min} and x_{max} are the minimum and maximum real values, respectively. To ensure an impartial comparison, the input construction was normalized for proposed LSTM model.

To illustrate the strengths and weaknesses of the proposed LSTM learning model, different parameterizations are used for training and prediction. Specifically, four maximum epochs numbers are tested, i.e., 50, 100, 150, and 200, in combination with different numbers of hidden nodes, i.e., 100 and 200 and different numbers of hidden layers, i.e., 1, 2 and 3.

In addition, the solver adopts adam optimization algorithm, and the learning rate is 0.005, dropout value is 0.2, and L2Regulation value is 0.01. Specific parameter configurations are described in the results section. Due to the influence of above optimized hyper-parameters on the prediction effect of different network architectures is relatively fixed, that is, it does not change with the number of network layers, the number of neurons and the number of epochs, the hyper-parameters are given directly without comparison.

The performance of manipulator inverse dynamics predictions is evaluated by root mean square error (RMSE), which is defined as follows.

$$RMSE = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}(p_i - y_i)^2} \tag{9}$$

In which, p_i and y_i represent the i-th predicted value and real value, respectively. N is the total number of the data.

B. RESULTS

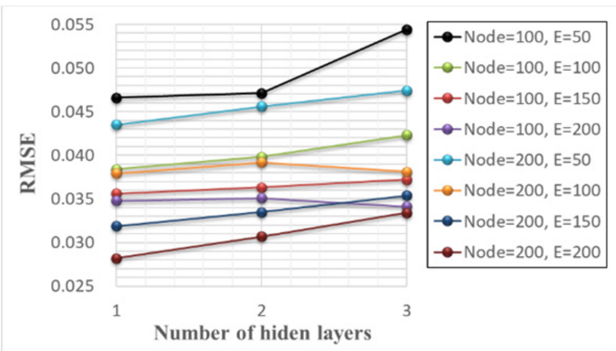
The training and prediction in this paper were performed with MATLAB 2019a, using ordinary personal computer. Computer hardware has a high influence on training time. For this paper, the models are trained on a CPU with a clock speed of 2.7 GHz

We compared the predicted joint torque of the robot with the simulation results based on different hyper-parameters of LSTM neural network. First, our results show that the three key hyper-parameters that affect the prediction accuracy are the number of hidden layers, number of neurons and number of maximum epochs. Furthermore, the increase of the maximum number of epochs can significantly improve the simulation accuracy, as shown in Figure 6 (this figure illustrates the influence of epochs number on the prediction accuracy of robot joint torque under different LSTM layers and different number of neurons. For example, "LSTM: 100" in the figure means that only one layer of LSTM is used and that the neuron number of the layer is 100; "LSTM: 66, 34" means that two layers of LSTM are used and that the first layer has 66 LSTM cells and the second layer has

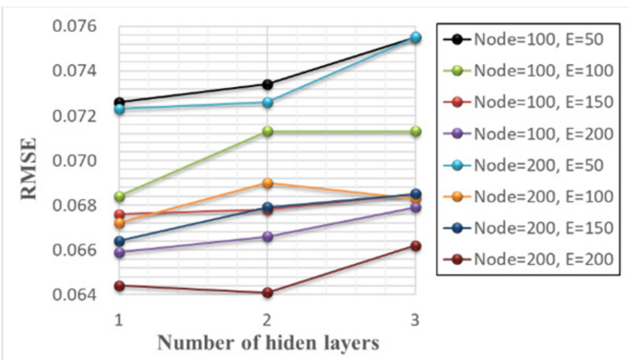
TABLE 2. Simulation results (RMSE) of different hyper-parameters and different input data.

	position $\boldsymbol{q}$, velocity $\dot{\boldsymbol{q}}$, acceleration $\ddot{\boldsymbol{q}}$				position $\boldsymbol{q}$			
	E=50	E=100	E=150	E=200	E=50	E=100	E=150	E=200
LSTM: 100	0.0466	0.0384	0.0356	0.0348	0.0726	0.0684	0.0676	0.0659
LSTM: 66, 34	0.0471	0.0398	0.0363	0.0351	0.0734	0.0713	0.0678	0.0666
LSTM: 58, 28, 14	0.0544	0.0423	0.0372	0.0341	0.0755	0.0713	0.0685	0.0679
LSTM: 200	0.0435	0.0379	0.0319	0.0282	0.0723	0.0672	0.0664	0.0644
LSTM: 133, L67	0.0456	0.0392	0.0335	0.0307	0.0726	0.0690	0.0679	0.0641
LSTM: 114, L57, L29	0.0474	0.0381	0.0354	0.0334	0.0755	0.0683	0.0685	0.0662

Description: this table illustrates the torque prediction results under different hyper-parameters, different input data and different network architectures. For example, the first torque prediction result "0.0466" is in such cases: the input data for is “position , velocity , acceleration ”, maximum epochs is 50, the network architecture is "LSTM: 100".

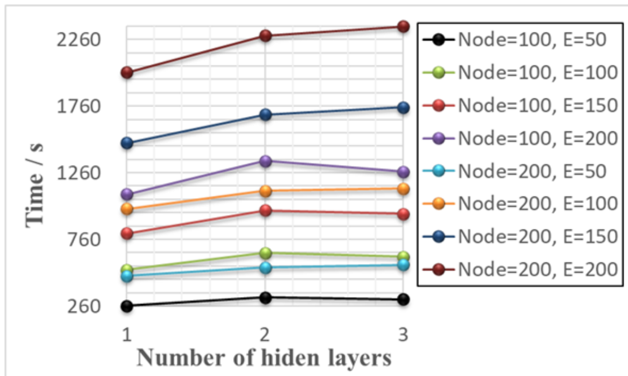


(a) Input: $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$

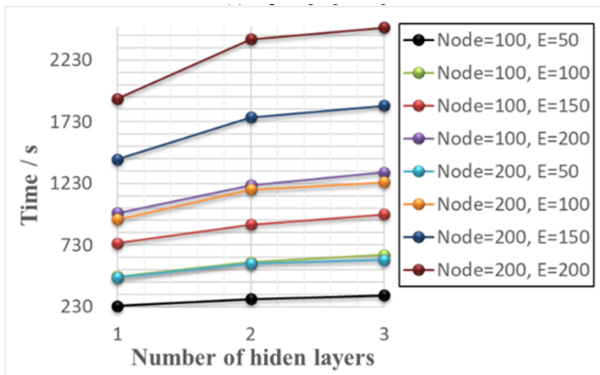


(b) Input: $\boldsymbol{q}$

FIGURE 8. The effect of number of hidden layers on prediction accuracy.



(a) Input: $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$



(b) Input: $\boldsymbol{q}$

FIGURE 9. The effect of number of hidden layers on training time.

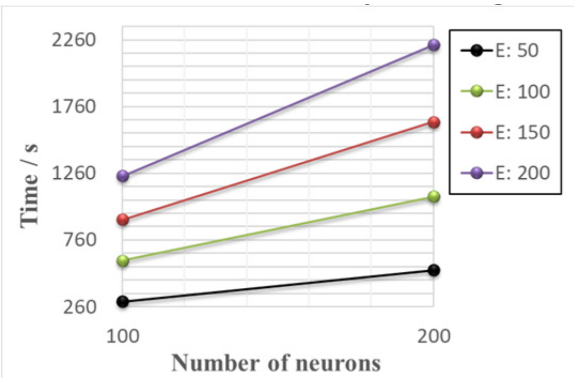
34 LSTM cells). Increasing the number of neurons can also improve the simulation accuracy, but the effect is not obvious, as shown in Figure 7 (this figure illustrates the influence of the number of neurons on the prediction accuracy of robot joint torque under different epochs number conditions. For example, “E: 50” in the figure means that the wpochs number is 50; “E: 100” means that the Epochs number is 100). It should also be noted that increasing the number of hidden layers in the neural network will reduce the prediction accuracy, as shown in Figure 8 (this figure illustrates the influence of LSTM layer number on the prediction accuracy of robot joint torque under different neuron number and epochs number conditions. For example, “Node=100, E=50” in the figure indicates that the number of neurons is 100 and the number of epochs is 50; The “Node=200, E=150” in the figure indicates that the number of neurons is 200 and the number of epochs is 150).

On the other hand, the predictive performance of all our simulation is shown in Table 2 and Table 3. The calculation results show that the best RMSE value obtained by the proposed LSTM model is 0.0282 in condition that the input data are position, velocity, acceleration. For the input data only position is 0.0641. Although the prediction accuracy of using only position as input data is lower than that of using position, velocity and acceleration at the same time, the calculation

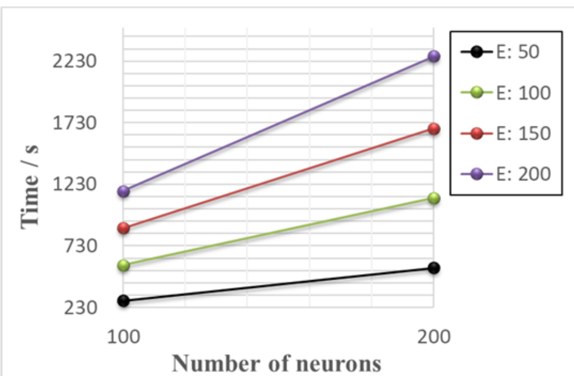
TABLE 3. Simulation results (Time/s) of different hyper-parameters and different input data.

	position $\boldsymbol{q}$, velocity $\dot{\boldsymbol{q}}$, acceleration $\ddot{\boldsymbol{q}}$				position $\boldsymbol{q}$			
	E=50	E=100	E=150	E=200	E=50	E=100	E=150	E=200
LSTM: 100	262	532	806	1098	236	475	745	989
LSTM: 66, 34	325	657	977	1349	292	591	897	1214
LSTM: 58, 28, 14	311	628	952	1269	322	649	979	1320
LSTM: 200	485	989	1485	2012	465	937	1426	1915
LSTM: 133, L67	548	1123	1696	2290	580	1180	1767	2400
LSTM: 114, L57, L29	568	1141	1750	2359	608	1240	1860	2493

Description: this table illustrates the torque prediction time consumption under different hyper-parameters, different input data and different network architectures. For example, the first torque prediction time consumption "262" is in such cases: the input data for is “position , velocity , acceleration ”, maximum epochs is 50, the network architecture is "LSTM: 100".c

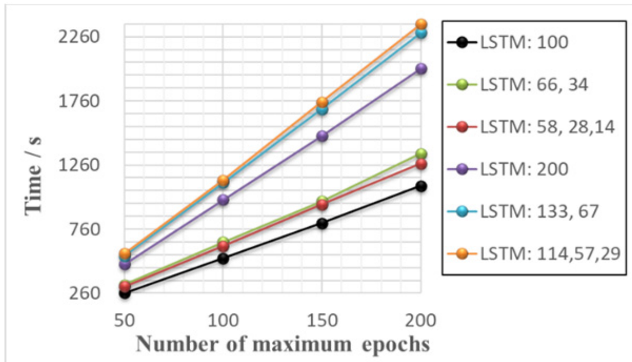


(a) Input: $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$

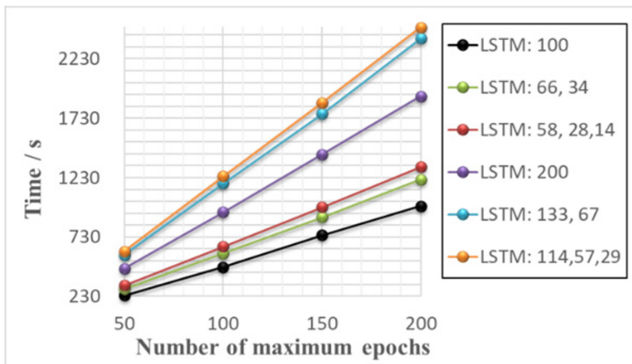


(b) Input: $\boldsymbol{q}$

FIGURE 10. The effect of number of neurons on training time.



(a) Input: $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$



(b) Input: $\boldsymbol{q}$

FIGURE 11. The effect of number of maximum epochs on training time.

of joint velocity and acceleration is omitted and the prediction performance is acceptable. In addition to the simulation accuracy, the calculation speed is also an important index to evaluate the performance of a learning model. In this paper, the time consumption is used as an evaluation index to compare the calculation speed of the different hyper-parameter settings and different input dataset. The simulation results show that there are significant differences in the calculation speed among the different hyper-parameter settings. However, different input data has little effect on training speed. In general, the training time consumption increases with the number of hidden layers (Figure 9), the number of neurons (Figure 10) and the number of epochs (Figure 11). Specifically, Figure 9 illustrates the influence of LSTM layer number on the training time under different neuron number and Epochs number conditions. For example, ‘‘Node=100, E=50’’ in the figure indicates that the number of neurons is 100 and the number of Epochs is 50; The ‘‘Node=200, E=150’’ in the figure indicates that the number of neurons is 200 and the number of Epochs is 150. Figure 10 illustrates the influence of the number of neurons on the training time under different epochs number conditions. For example, ‘‘E: 50’’ in the figure means that the Epochs number is 50;‘‘E: 100’’ means that the Epochs number is 100. Figure 11 illustrates the influence of epochs number on the training time under different LSTM layers and different number of neurons. For example, "LSTM: 100" in the figure means that only one layer of LSTM is used, and the neuron number of the layer is 100; "LSTM: 66, 34" means that two layers of LSTM are used. The first layer has 66 LSTM cells and the second layer has 34 LSTM cells.)

IV. CONCLUSION

The accuracy of inverse dynamics prediction is very important for robot control for smart cities and factories. At present, according to the theoretical basis of the model, manipulator inverse dynamics models are divided into two main categories: models based on physical concepts and AI models techniques. However, in practice, the inverse dynamics is affected by many uncertain factors, such as flexibility of connecting rods and joints, joint clearance and friction, etc., which limits the application of physical models. AI models, or data-driven models, are able to autonomously learn the uncertain factors from the actual data of manipulator.

With the development of smart cities and factories technology, the applications of machine learning, especially deep learning, have also expanded. This paper proposed a deep learning model architecture based on LSTM to predict the manipulator inverse dynamics. Detailed discussion and recommendation are made with respect to the process of model parameter settings, simulation performances, and applications under different input data. The main conclusions are as follows: with respect to parameter setting, our results show the effects of the number of maximum epochs on model performance should be prioritized. The effects of the number of hidden nodes on model performance are limited. While increasing the number of hidden layers in the neural network will reduce the prediction accuracy. Meanwhile, in the process of model construction, due to differences in data volume and structure, model parameters have a different influence on model performance. Therefore, we suggest that the model should be repeatedly trained before practical application to determine the optimal parameters and ensure the prediction ability of the model. Using the LSTM-based deep learning model proposed in this paper, joint torque prediction can not only use the joint position, velocity and acceleration of the manipulator, but also only use the joint position as input data. Although the prediction accuracy of using only position as input data is lower than that of using position, velocity and acceleration at the same time, the calculation of joint velocity and acceleration is omitted and the prediction performance is acceptable. The limitation of this work is that the torque prediction precision is not good enough only using joint position as input data. Therefore, in the future, further research can be carried out on this problem. The training time consumption of the proposed learning model increases with the number of hidden layers, the number of neurons and the number of epochs. Although increasing the number of epochs will increase the training time, it is necessary. Therefore, we suggest that the maximum number of epochs should be as large as possible to ensure the accuracy of joint torque prediction. While increasing the number of neurons is not very obvious for improving the accuracy of joint torque prediction, but the training time increases a lot. Therefore, it is not recommended to use too many neurons to improve the prediction accuracy.

REFERENCES

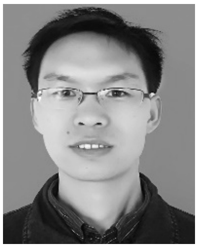
[1] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: A survey," *Cogn. Process.*, vol. 12, no. 4, pp. 319–340, 2011.
[2] S. P. Chan, "Real time control of robot manipulator using a neural network based learning controller," in *Proc. 19th Annu. Conf. IEEE Ind. Electron.*, vol. 3, Nov. 1993, pp. 1825–1830.
[3] Z. Su and K. Khorasani, "A neural-network-based controller for a single-link flexible manipulator using the inverse dynamics approach," *IEEE Trans. Ind. Electron.*, vol. 48, no. 6, pp. 1074–1086, Dec. 2001.
[4] W. Chatlatanagulchai, H. C. Nho, and P. H. Meckl, "Robust observer backstepping neural network control of flexible-joint manipulator," in *Proc. Amer. Control Conf.*, vol. 6, Jun./Jul. 2004, pp. 5250–5255.
[5] W. Chatlatanagulchai and P. H. Meckl, "Motion control of two-link flexible-joint robot, using backstepping, neural networks, and indirect method," in *Proc. IEEE Conf. Control Appl.*, vol. 1, Aug. 2005, pp. 601–605.
[6] M. H. Shaheed, H. Poerwanto, and M. O. Tokhi, "Adaptive inverse-dynamic and neuro-inverse-dynamic active vibration control of a single-link flexible manipulator," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 219, no. 6, pp. 431–448, 2005.
[7] N. Luan, "Study on robot inverse dynamics modeling with neural network," *Electr. Mach. Control*, vol. 12, no. 2, pp. 174–178, 2008.
[8] H. Mori, Y. Ohama, and N. Fukumura, "Learning of real robot's inverse dynamics by a forward-propagation learning rule," *Elect. Eng. Jpn.*, vol. 161, no. 4, pp. 38–48, 2007.
[9] Z.-H. Jiang, T. Ishida, and M. Sunawada, "Neural network aided dynamic parameter identification of robot manipulators," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2006, pp. 3298–3303.
[10] A. F. R. Araujo and H. D'Arbo, "A partially recurrent neural network to perform trajectory planning, inverse kinematics, and inverse dynamics," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 2, Oct. 1998, pp. 1784–1789.
[11] K. Onozato and Y. Maeda, "Learning of inverse-dynamics and inverse-kinematics for two-link scara robot using neural networks," in *Proc. SICE Annu. Conf.*, Sep. 2007, pp. 1031–1034.
[12] N. Ishibashi and Y. Maeda, "Learning of inverse-dynamics for SCARA robot," in *Proc. SICE Annu. Conf.*, Sep. 2011, pp. 1300–1303.
[13] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local Gaussian process regression," *Adv. Robot.*, vol. 23, no. 15, pp. 2015–2034, 2009.
[14] D. Nguyen-Tuong and J. Peters, "Using model knowledge for learning inverse dynamics," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2010, pp. 2677–2682.
[15] J. S. de la Cruz, "Learning inverse dynamics for robot manipulator control," M.S. thesis, Dept. Elect. Comput. Eng., Univ. Waterloo, Waterloo, ON, Canada, 2011.
[16] J. S. de la Cruz, W. Owen, and D. Kulíc, "Online learning of inverse dynamics via Gaussian process regression," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 3583–3590.
[17] R. Camoriano, S. Traversaro, L. Rosasco, G. Metta, and F. Nori, "Incremental semiparametric inverse dynamics learning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 544–550.
[18] C.-A. Cheng, H.-P. Huang, H.-K. Hsu, W.-Z. Lai, and C.-C. Cheng, "Learning the inverse dynamics of robotic manipulators in structured reproducing Kernel Hilbert space," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1691–1703, Jul. 2016.
[19] D. Romeres, M. Zorzi, R. Camoriano, and A. Chiuso, "Online semi-parametric learning for inverse dynamics modeling," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 2945–2950.
[20] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, Jun. 2019.

[21] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: A path based deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 100, pp. 372–385, Mar. 2019.
[22] W. Long, Z. Lu, and L. Cui, "Deep learning-based feature engineering for stock price movement prediction," *Knowl.-Based Syst.*, vol. 164, pp. 163–173, Jan. 2019.
[23] Z. Cen and J. Wang, "Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer," *Energy*, vol. 169, pp. 160–171, Feb. 2018.
[24] H. Yan and H. Ouyang, "Financial time series prediction based on deep learning," *Wireless Pers. Commun.*, vol. 102, no. 2, pp. 683–700, 2018.
[25] S. Jha, R. Kumar, L. H. Son, M. Abdel-Basset, I. Priyadarshini, R. Sharma, and H. V. Long, "Deep learning approach for software maintainability metrics prediction," *IEEE Access*, vol. 7, pp. 61840–61855, 2019.
[26] T. T. Dao, "From deep learning to transfer learning for the prediction of skeletal muscle forces," *Med. Biol. Eng. Comput.*, vol. 57, no. 5, pp. 1049–1058, 2019.
[27] S. Cai, A. Palazoglu, L. Zhang, and J. Hu, "Process alarm prediction using deep learning and word embedding methods," *ISA Trans.*, vol. 85, pp. 274–283, Feb. 2019.
[28] C. Xu, H. Chen, J. Wang, Y. Guo, and Y. Yuan, "Improving prediction performance for indoor temperature in public buildings based on a novel deep learning method," *Building Environ.*, vol. 148, pp. 128–135, Jan. 2019.
[29] W. Wang, J. Shen, J. Xie, M.-M. Cheng, H. Ling, and A. Borji, "Revisiting video saliency prediction in the deep learning era," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[30] K. M. Tsiouris, V. C. Pezoulas, M. Zervakis, S. Konitsiotis, D. D. Koutsouris, and D. I. Fotiadis, "A long short-term memory deep learning network for the prediction of epileptic seizures using EEG signals," *Comput. Biol. Med.*, vol. 99, pp. 24–37, Aug. 2018.
[31] C. Tong, X. Yin, S. Wang, and Z. Zheng, "A novel deep learning method for aircraft landing speed prediction based on cloud-based sensor data," *Future Gener. Comput. Syst.*, vol. 88, pp. 552–558, Nov. 2018.
[32] P.-W. Soh, J.-W. Chang, and J.-W. Huang, "Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations," *IEEE Access*, vol. 6, pp. 38186–38199, 2018.
[33] S. Singaravel, J. Suykens, and P. Geyer, "Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction," *Adv. Eng. Inform.*, vol. 38, pp. 81–90, Oct. 2018.
[34] H. Liu, X. Mi, and Y. Li, "Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network," *Energy Convers. Manage.*, vol. 166, pp. 120–131, Jun. 2018.
[35] A. S. Polydoros, E. Boukas, and L. Nalpantidis, "Online multi-target learning of inverse dynamics models for computed-torque control of compliant manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4716–4722.
[36] A. S. Polydoros, L. Nalpantidis, and V. Krüger, "Real-time deep learning of robotic manipulator inverse dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 3442–3448.
[37] B. Liang, T. Li, Z. Chen, Y. Wang, and Y. Liao , "Robot arm dynamics control based on deep learning and physical simulation," in *Proc. 37th Chin. Control Conf. (CCC)*, 2018, pp. 2921–2925.
[38] E. Rueckert, M. Nakatenus, S. Tosatto, and J. Peters, "Learning inverse dynamics models in O(n) time with LSTM networks," in *Proc. IEEE-RAS 17th Int. Conf. Hum. Robot.*, Nov. 2017, pp. 811–816.
[39] Q. Yong, "Information potential fields navigation in wireless Ad-Hoc sensor networks," *Sensors*, vol. 11, no. 5, pp. 4794–4807, 2011.
[40] W. Wei, L. Wang, X. H. Hei, P. Shen, W. Shi, and L. Shan, "GI/Geom/1 queue based on communication model for mesh networks," *Int. J. Commun. Syst.*, vol. 27, no. 11, pp. 3013–3029, Nov. 2014.
[41] W. Wei, X. L. Yang, P. Y. Shen, and B. Holes, "Holes detection in anisotropic sensornets: Topological methods," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 10, Oct. 2012, Art. no. 135054.

NAN LIU was born in Tianjin, China, in 1986. He received the B.Tech. and M.Tech. degrees in mechatronic engineering from the Tianjin University of Technology, in 2009 and 2012, respectively. He is currently pursuing the Ph.D. degree with Tianjin Polytechnic University. He has been a Lecturer with the Tianjin University of Technology, since 2012. He has published around ten research articles in international journals and conferences. His current research interest includes robotic dynamics and intelligent control for smart cities and factories. In addition, his current research interest includes wave theory and technology.

LIANGYU LI received the B.Sc. degree from the Taiyuan University of Technology, in 1987, and the M.Sc. and Ph.D. degrees from the Harbin Institute of Technology, in 1992 and 2000, respectively. He is currently a Professor with the Department of Mechanical Engineering, Tianjin Polytechnic University. He has published over 40 journal and conference papers. His current research interests include industrial robot technology and welding automation.

BING HAO received the B.Tech. degree from Chongqing University, in 1988. He is currently the Director of the State Key Laboratory of Heavy Mining Equipment (CITICHIC). He is also the Deputy General Manager of CITIC Heavy Industries Company, Ltd. He presided over or participated in nearly 20 national and provincial scientific research projects and published over ten articles. He holds more than 30 national authorized patents. He presided over more than ten major equipment, such as the national strategic innovation product large high-pressure roller mill, the first 10-million-ton semi-mobile crushing station in China, and the first liner plate replacement mechanical arm of large mill in China. He received more than ten national and provincial awards. He is the Vice President of Mining Machinery Branch, China Heavy Machinery Industry Association.



LIUSONG YANG received the M.Tech. degree from the Henan University of Science and Technology, in 2007. He is currently the Director of the Simulation Analysis and Industrial Design Center, CITIC Heavy Industries Company, Ltd. He presided over or participated in two provincial-level scientific research projects and published nearly 20 articles. He has more than ten authorized invention patents.

TONGHAI HU received the B.Tech. degree from the Changchun University of Technology, in 1988. He is currently a Technical Expert of CITIC Heavy Industries Company, Ltd. He has presided over one National Key Research and Development Program. He presided over three major industrialization projects, such as development, design, and industrial application of replacement manipulator for liner plate for large-scale grinding. He was granted more than ten invention patents and published five articles. He received two provincial and ministerial awards.

TAO XUE received the Ph.D. degree from Yanshan University, in 2014. He joined the Tianjin University of Technology, in 2016, where he is currently a Lecturer with the School of Mechanical Engineering. He has published several articles in major robotics journals and conferences. He has led several research projects. His current research interest includes dynamics and intelligent control for heavy robotic systems. He is a member of CMES.

SHOUJUN WANG received the M.Tech. degree in hydraulics and pneumatics from Zhejiang University, in 1989. He is currently a Professor and the Dean of the Mechanical Engineering College, Tianjin University of Technology. He has published over 100 articles. His current research interests include fluid transmission and control, electromechanical system integration, and unit technology. He led more than 60 research projects sponsored by the Ministry of Transport, the Ministry of Science and Technology, and universities and research institutes, to develop intelligent robots and marine power environment generation technologies. He is a Senior Member of the Chinese Mechanical Engineering Society and the Executive Director of the China Marine Engineering Society.