Research Article

The Use of Deep Learning Model for Effect Analysis of Conventional Friction Power Confinement

Chuntong Liu, Xin Wang, and Zhenxin He

Xi’an Research Institute of High Technology, Baqiao District, Tongxin Road, Xi’an City, Shaanxi 710025, China

Correspondence should be addressed to Zhenxin He; 201701350129@lzpcc.edu.cn

Received 6 April 2022; Revised 24 April 2022; Accepted 28 April 2022; Published 7 June 2022

Academic Editor: Naeem Jan

Copyright © 2022 Chuntong Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nonlinear friction could affect the high-precision motion system, resulting in poor tracking accuracy in the end. This is due to the fact that the Lugre friction model’s parameter identification process comprises both static and dynamic parameter identification. The convolutional neural network (CNN) model is used in this study to create the friction identification system. We suggest a hybrid methodology that combines the CNN method and the classic least-squares technique. The convolutional layer (CONV), which is defined by a convolutional kernel, analyzes and extracts features from an input image. In terms of accuracy and convergence, the results reveal that the upgraded CNN friction model outperforms the original CNN friction model. You may successfully reduce the influence of friction on your system while improving its performance by applying the feedforward correction.

1. Introduction

The industrial robot has become an indispensable automation tool in modern human society. Improving its control accuracy has always been a research hotspot at home and abroad. The traditional PID control has been unable to meet the accuracy requirements of the actual work, and the model-based control method has become the mainstream. The model-based controller needs to take the robot’s dynamic parameters as a priori value [1], but the robot is a multivariable and strongly coupled nonlinear system [2]. It is difficult to obtain the dynamic model through a mathematical calculation, and the experimental identification method is generally used. In the process of moving, the machine will be disturbed by nonlinear friction [3]. This interference is local and has high frequency and huge amplitude, and it has a significant impact on the system’s local performance. Due to the randomness of its position, the tracking accuracy of the whole travel range is reduced [4]. For this phenomenon, the corresponding friction model is used to identify the relevant parameters of the friction model through the parameter identification method, compensate for the nonlinear friction in the system, reduce the influence of the nonlinear friction in the system on the high-precision motion of the system, and improve the local tracking accuracy [5, 6].

The Lugre friction model is a typical friction model for servo systems and can accurately describe the frictional characteristics of the system during motion [7]. As the parameter identification of the Lugre friction model involves both static and dynamic parameter identification [8], it is a harmonious combination of static and kinematic characteristics, and the improved genetic algorithm can effectively improve the accuracy of identification by taking into account both static characteristics and dynamic factors and prevent the problem of falling into local optimality instead of global optimality in the process of identification [9–12]. This study first defines and introduces the Lugre friction model, then moves on to the improved CNN model’s implementation approach and the static parameter recognition procedures [13]. The method is confirmed by first collecting relevant position and drive force data in a loop system using a cylindrical linear motor based on dSPACE hardware using constant velocity trials, calculating the velocity and friction data values according to the basic physical equations, identifying the friction model using the basic and improved CNN models, and designing a feedforward compensation.
controller based on the identified friction model to achieve the nonlinear friction compensation [14].

The paper’s section-wise paragraph is as follows: The related work is presented in Section 2. Section 3 analyzes the Lugre friction model description and introduction. Section 4 describes the building convolutional neural network models. Section 5 discusses the experimental verifications. Finally, in Section 6, the research work is concluded.

2. Related Work

Many robot parameter identification methods have been proposed by researchers at home and abroad [15, 16]. A method for deriving the minimum set of parameters for tandem robots was proposed, which can reduce the number of operations for identification and improve the robustness of the algorithm. In [17], a recursive least-squares method was used to perform parameter operations, which improved the efficiency of the algorithm. [18] proposed a distribution identification method to reduce the complexity of the identification equations. The progress of robot dynamic parameter identification has been aided by intelligent control methods. [19] used an artificial bee colony algorithm, [20–22] used a particle swarm algorithm, and [23] used an improved genetic algorithm for identification, all of which achieved good identification results.

In order to improve the kinetic model recognition accuracy of industrial robots, this paper proposes a kinetic model recognition method based on an artificial neural network in combination with machine learning and deep learning algorithms that have emerged in recent years [24, 25].

3. Lugre Friction Model Description and Introduction

In 1995, scholars proposed the Lugre model [14], which is represented by the fact that the contact surfaces of two objects are in microscopic contact through elastic bristles, and when they are displaced by mutual tangential forces, the bristles of the contact surfaces will undergo elastic deformation, generating friction in the process, as shown in Figure 1.

The friction of the servo system can be expressed by the differential equation as

$$M \frac{d^2x}{dt^2} = F - F_f,$$

where $M$ represents the mass of the load, $x$ represents the displacement of the mass of the load, $F$ represents the driving force of the motor, and $F_f$ represents the frictional force.

The frictional force $F_f$ can be determined by the following mathematical formula:

$$\dot{z} = v - \frac{\sigma_0}{g(v)} z|v|,$$

$$g(v) = F_c + (F_s - F_c)e^{-(v/v_s)^2},$$

$$F_f = \sigma_0 \dot{z} + \sigma_1 z + \sigma_2 v.$$  

When the system is in a steady state, at this time, the system $dx/dt = a$, where $a$ is a constant, i.e., at this time $z = 0$, then from equation (5), we have

$$z = \frac{g(v)}{\sigma_0} v = \frac{g(v)}{\sigma_0} \text{sgn} (v),$$

where the $\text{sgn}$ function is a symbolic function, substituting equations (5) and (3) into equation (4) is the frictional force when the system is in a steady state:

$$F_f = \left(F_c + (F_s - F_c)e^{-(v/v_s)^2}\right) \text{sgn} (v) + \sigma_2 v.$$  

4. Building Convolutional Neural Network Models

CNN is a type of neural network that may be used to classify images. CNN merely connects the nodes between two adjacent layers and shares the weights, unlike earlier fully connected neural networks. This optimizes the neural network, reduces the model’s complexity, and enhances the operation’s efficiency greatly.

The convolutional layer (CONV), which is defined by a convolutional kernel, performs feature extraction on the input image. The convolution is expressed as $f(x) = wx + b$. The convolution kernel is also known as a filter or “field of perception.” The convolution kernel is weighted and processed by a weight matrix with the local data of the input image, and the kernel slides over the input data to extract features from the whole image. The convolutional feature map is calculated as

$$w_{out} = \frac{w_{in} - F + 2sP}{S} + 1,$$

where $w_{out}$ is the output feature map size, $w_{in}$ is the input feature map size, $F$ is the convolutional kernel size, and $S$ is the convolutional step size.

The pooling layer (POOL), reduces the spatial dimension of the picture, keeps the depth constant, reduces the network connection parameters, optimizes computational efficiency, and prevents overfitting.

The fully connected layer (FC), where the previous layers have been completed with highly abstract information features, is used for classification through the fully connected layer. The last convolutional layer needs to be matrix flattened when connected to the fully connected layer.
The Softmax layer uses Softmax to classify the linear output. Assuming the original array $V$, $V_i$ is the $i$th element of the array, then $V_i$ of Softmax($\mathbf{\hat{y}}$) = $e^i / \sum_{i=1}^{n} e^i$.

The structural design of the CNN-based neural network for parameter identification of the sassafras model includes 11 hidden layers, 4 convolutional layers, 4 pooling layers, and 3 fully connected layers, and the specific parameters are shown in Table 1.

(1) **Input Layer.** The input layer collects the image dataset of the quality inspection agency for the identification of the parameters of the sassafras model. The image dataset is preprocessed to remove noise and delete poor-quality data. The image preprocessing is in RGB (100 × 100 × 3) format. The ratio = 0.8 was set to split the training and validation sets, with 80% of the data used for model training and 20% for model validation.

(2) **Convolutional Layers.** The model is designed with 4 convolutional layers, the sizes of which are $5 \times 5 \times 3$, $5 \times 5 \times 32$, $3 \times 3 \times 64$, and $3 \times 3 \times 128$, all with a step size of 1. The first two layers are designed with larger sizes in order to expand the perceptual field in the first stage, extract more image features, and reduce the number of computation layers. The last two layers are smaller in order to increase the depth of the model, reduce the number of computational parameters, and reduce computational complexity.

(3) **Pooling Layers.** Four pooling layers are designed to correspond to 4 convolutional layers, all of size $2 \times 2$, with depth corresponding to the previous convolutional layer. After pooling, the image data can be reduced in feature dimension and number of parameters to reduce overfitting.

(4) **Fully Connected Layers.** Three fully connected layers are designed to flatten the convolutional pooled array matrix into a column vector. The final output is a parametric identification classification of the 5 levels of the sassafras model. The fully connected is chosen to reduce overfitting by dropout, L2 regular terms, etc.

---

**Table 1: Structure of neural network for parameter identification based on CNN sassafras model.**

| Layered     | Convolution layer (1 layer) | Convolution layer (2 layers) | Convolution layer (3 layers) | Convolution layer (4 layers) | Convolution layer (5 layers) | Convolution layer (6 layers) | Convolution layer (7 layers) | Convolution layer (8 layers) |
|-------------|----------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|
| Nuclear size| $5 \times 5 \times 3$      | $2 \times 2 \times 3$        | $5 \times 5 \times 32$       | $2 \times 2 \times 32$       | $3 \times 3 \times 64$       | $2 \times 2 \times 64$       | $3 \times 3 \times 128$      | $2 \times 2 \times 128$      |
| Step        | 1                          | 2                            | 1                            | 2                            | 1                            | 2                            | 1                            | 2                            |
| Layered     | Full connection layer (9 layers) | Full connection layer (10 layers) | Full connection layer (11 layers) | Output | Activation function | --- | --- | --- |
| Size        | $6 \times 6 \times 128$ | $1024 \times 1$ | $512 \times 1$ | 5 | Classification function | Softmax |
| Overfitting | Dropout                    | Dropout                      | L2                           | ---                           | ---                           | ---                           | ---                           | ---                           |

---

![6 DOF robot arm](image)

---

Figure 2: Six-degree-of-freedom robotic arm.

Arm joints are connected by bearings

Stainless steel hexagonal screws

---

Metal robot arm frame
4 x 1501 digital servo
2 x 1501 metal micro servo

---

6 DOF robot arm
(5) *Activation Function.* Choose ReLU as the activation function; the function of the data processing has the characteristics of fast convergence and rapid gradient reduction.

(6) *Classification Function.* Softmax is selected as the classification function, which will classify the input textile photos into five corresponding class categories, so as to achieve the goal of intelligent assessment of textile hairball.

5. **Experimental Verification**

In this chapter, we defined the experimental design, identification results, and analysis of results in detail.

5.1. **Experimental Design.** A 6-degree-of-freedom robotic arm produced by a company is shown in Figure 2. The artificial neural network model presented in Sections 1 and 2 of this paper was built using the deep learning framework.
Keras, with a dropout regularization layer added to avoid overfitting [26–30].

The more training data provided, the more features of the model are theoretically identified, and the greater generalization capacity the model will have while performing neural network identification. Using the stimulated trajectory design approach described in Section 2, we collected as much actual data from the robotic arm as possible and separated it into three parts: the training set (70%), the validation set (15%), and the test set (10%) (15 percent). Figures 3 and 4 demonstrate how the magnitude of the loss function and the training accuracy were assessed during the training process. It can be seen that as the iterations progress, the loss function of the model gradually decreases close to zero and the accuracy gradually reaches its maximum value, reaching convergence at around the 100th round, and the accuracy no longer changes.

The use of machine learning and deep learning methods for solving similar nature problems has extensively been studied by the research community [31–33].

5.2. Identification Results. The traditional least-squares method paired with the Coulomb viscous friction model was used to identify the same experimental data and
validated on the same validation set in order to test the accuracy of the algorithm suggested in this paper. The moment curves obtained by the 2 algorithms were put together, and the results are shown in Figure 5.

From Figure 5, it can be seen that both algorithms have a good following effect on the actual torque. By looking at the local magnification, we can see that the neural network algorithm proposed in this paper has a better fit to the actual torque, especially around the peak, and has a smoother curve for the actual torque, which is more suitable for the controller design. The error curve of joint 1 is shown in Figure 6, which shows that the torque error calculated by the neural network recognition algorithm is smaller than that of the traditional algorithm, with a smaller peak error and a smoother curve. It is worth noting that there are some points in the curve with large jumps in error, mainly around the point where the joint speed of the robot arm is 0. The relative error is calculated with a small denominator, resulting in a larger relative error result, which has no impact on practical use.

As shown in equation (8), the matching degree $\delta$ for kinetic discrimination can be calculated using the calculated moment $\tau_i$ and the experimentally measured moment $\tau$:

$$\delta = \left(1 - \frac{||\tau_i - \tau||}{||\tau_i||}\right) \times 100\%.$$  \hspace{1cm} (8)

The matching results of the 2 methods are shown in Table 2. As can be seen from Table 1, the matching degree of the neural network model can be improved by more than 5% on average compared to the traditional least-squares model.

In recent years, various optimization algorithms have been increasingly used in the field of robot identification; for example, the artificial bee colony algorithm combined with the nonlinear friction model has achieved good experimental results with an average accuracy of 89%. The average accuracy of the algorithm proposed in this paper is 90.70%, which is slightly better than the artificial swarm algorithm.

5.3. Analysis of Results. When compared to the traditional least-squares identification approach, the accuracy of the identification method described in this study is improved, and the torque fluctuations of the traditional identification model are successfully minimised and smoothed out. The method described in this paper is more suitable for controller design since frequent torque fluctuations can easily damage the controller. The combination of a neural network model and a traditional least-squares model can be used in a practical control system to obtain better results.

Frictional forces in robots are related to several external factors, and modeling and identification of frictional forces have been a major challenge. In recent years, many algorithms (e.g., artificial bee colony algorithms) have been
proposed whose recognition accuracy depends on the choice of the friction model, and the friction models of different robots in different working environments are not identical. The method proposed in this paper does not require special modeling of friction, is highly adaptable and portable for robots in different working environments, and ensures high accuracy. Therefore, the method proposed in this paper also has certain advantages over the more recent identification methods currently available.

6. Conclusions

To address the problem of low accuracy of classic kinetic parameter identification techniques, we present an artificial neural network-based identification method employing the ReLU activation function in combination with the RMSProp algorithm and the dropout method in this research. Experiments are being conducted to see how well they compare to the classic least-squares method. The results show that the proposed method can significantly improve the smoothness of the moment calculation, and the accuracy is improved by more than 5% compared to the traditional method. The algorithm does not require modeling of the system friction and has good adaptability.

Data Availability

The datasets used during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflict of interest.

References

[1] J. Zhao and F. Wang, "Parameter identification by neural network for intelligent deep drawing of axisymmetric work-pieces," Journal of Materials Processing Technology, vol. 166, no. 3, pp. 387–391, 2005.
[2] J. Jia, M. Zhang, X. Zang, H. Zhang, and J. Zhao, "Dynamic parameter identification for a manipulator with joint torque sensors based on an improved experimental design," Sensors, vol. 19, no. 10, p. 2248, 2019.
[3] N. Abas, A. Legowo, and R. Akmeliawati, "Parameter identification of an autonomous quadrotor," in 2011 4th international conference on mechatronics (ICOMM), pp. 1–8, IEEE, 2011.
[4] F. Alonge, M. Cirrincione, F. D’ippolito, M. Pucci, and A. Sferlazza, "Parameter identification of linear induction motor model in extended range of operation by means of input-output data," IEEE Transactions on Industry Applications, vol. 50, no. 2, pp. 959–972, 2014.
[5] F. Bu, Z. Ma, Y. Yuan, and Z. Wang, "WECC composite load model parameter identification using evolutionary deep reinforcement learning," IEEE Transactions on Smart Grid, vol. 11, no. 6, pp. 5407–5417, 2020.
[6] M. Nolte, N. Kister, and M. Maurer, "Assessment of deep convolutional neural networks for road surface classification," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 381–386, IEEE, 2018.
[7] C. Rechea, S. Levasseur, and R. Finno, "Inverse analysis techniques for parameter identification in simulation of excavation support systems," Computers and Geotechnics, vol. 35, no. 3, pp. 331–345, 2008.
[8] E. Šabanović, V. Žurulais, O. Prentkovskis, and V. Skrickij, "Identification of road-surface type using deep neural networks for friction coefficient estimation," Sensors, vol. 20, no. 3, p. 612, 2020.
[9] G. Yang, Q. J. Li, Y. Zhan, Y. Fei, and A. Zhang, "Convolutional neural network–based friction model using pavement texture data," Journal of Computing in Civil Engineering, vol. 32, no. 6, p. 04018052, 2018.
[10] X. Tao, Z. Chunjiuon, and X. Yongjian, "Collaborative parameter update based on average variance reduction of historical gradients [1]," Journal of Electronics and Information Technology, vol. 43, no. 4, pp. 956–964, 2021.
[11] Z. H. A. N. G. Zhengwan, Z. H. A. N. G. Chunjiong, L. I. Hongbing, and X. I. E. Tao, "Multipath transmission selection algorithm based on immune connectivity model," Journal of Computer Applications, vol. 40, no. 12, p. 3571, 2020.
[12] Z. Zhang, D. Wu, and C. J. Zhang, "Study of cellular traffic prediction based on multi-channel sparse LSTM [1]," Computer Science, vol. 48, no. 6, pp. 296–300, 2021.
[13] J. Zhang and P. Xia, "An improved PSO algorithm for parameter identification of nonlinear dynamic hysteretic models," Journal of Sound and Vibration, vol. 389, pp. 153–167, 2017.
[14] Z. H. Huang, L. I. Zhang, S. Y. Cheng, J. Zhang, and X. H. Xia, "Back-analysis and parameter identification for deep excavation based on Pareto multiobjective optimization," Journal of Aerospace Engineering, vol. 28, no. 6, p. A4014007, 2015.
[15] R. Hasanpour, J. Rostami, J. Schmitt, Y. Ozcelik, and B. Sohrabian, "Prediction of TBM jamming risk in squeezing grounds using Bayesian and artificial neural networks," Journal of Rock Mechanics and Geotechnical Engineering, vol. 12, no. 1, pp. 21–31, 2020.
[16] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S. Sukhatme, "NeuralSim: augmenting differentiable simulators with neural networks," in 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 9474–9481, IEEE, 2021.
[17] A. Zirpoli, G. Maier, G. Novati, and T. Garbowski, "Dilatometric tests combined with computer simulations and parameter identification for in-depth diagnostic analysis of concrete dams," in Life-Cycle Civil Engineering, pp. 279–284, CRC press, 2008.
[18] J. P. Ponthot and J. P. Kleinermann, "A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming simulation," Computer Methods in Applied Mechanics and Engineering, vol. 195, no. 41-43, pp. 5472–5508, 2006.
[19] S. Vardakos, M. Gutierrez, and C. Xia, "Parameter identification in numerical modeling of tunneling using the Differential Evolution Genetic Algorithm (DEGA)," Tunnelling and Underground Space Technology, vol. 26, pp. 109–123, 2012.
[20] F. Aghazadeh, A. Tahan, and M. Thomas, "Tool condition monitoring using spectral subtraction and convolutional neural networks in milling process," The International Journal of Advanced Manufacturing Technology, vol. 98, no. 9-12, pp. 3217–3227, 2018.
[21] M. Abendroth and M. Kuna, "Identification of ductile damage and fracture parameters from the small punch test using...
neural networks,” *Engineering Fracture Mechanics*, vol. 73, no. 6, pp. 710–725, 2006.

[22] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, “Prediction of aerodynamic flow fields using convolutional neural networks,” *Computational Mechanics*, vol. 64, no. 2, pp. 525–545, 2019.

[23] M. Abendroth and M. Kuna, “Determination of deformation and failure properties of ductile materials by means of the small punch test and neural networks,” *Computational Materials Science*, vol. 28, no. 3-4, pp. 633–644, 2003.

[24] D. Zhou, Q. Yao, H. Wu, S. Ma, and H. Zhang, “Fault diagnosis of gas turbine based on partly interpretable convolutional neural networks,” *Energy*, vol. 200, article 117467, 2020.

[25] B. Lecampion, A. Constantinescu, and D. Nguyen Minh, “Parameter identification for lined tunnels in a viscoplastic medium,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 26, no. 12, pp. 1191–1211, 2002.

[26] F. Ullah, F. Al-Turjman, and A. Nayyar, “IoT-based green city architecture using secured and sustainable android services,” *Environmental Technology and Innovation*, vol. 20, article 101091, 2020.

[27] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “Generative and discriminative voxel modeling with convolutional neural networks,” 2016, https://arxiv.org/abs/1608.04236.

[28] Z. Liu, H. Fang, and J. Xu, “Identification of piecewise linear dynamical systems using physically-interpretable neural-fuzzy networks: methods and applications to origami structures,” *Neural Networks*, vol. 116, pp. 74–87, 2019.

[29] M. Lutter, K. Listmann, and J. Peters, “Deep Lagrangian networks for end-to-end learning of energy-based control for under-actuated systems,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7718–7725, IEEE, 2019.

[30] F. Ullah, M. R. Naeem, H. Naeem, X. Cheng, and M. Alazab, “CroLSSim: cross-language software similarity detector using hybrid approach of LSA-based AST-MDrep features and CNN-LSTM model,” *International Journal of Intelligent Systems*, 2022.

[31] R. Ali, A. Ali, F. Iqbal, A. M. Khattak, and S. Aleem, “A systematic review of artificial intelligence and machine learning techniques for cyber security,” in *International Conference on Big Data and Security*, pp. 584–593, Springer, Singapore, 2020.

[32] R. Ali, M. H. Siddiqi, and S. Lee, “Rough set-based approaches for discretization: a compact review,” *Artificial Intelligence Review*, vol. 44, no. 2, pp. 235–263, 2015.

[33] R. Ali, S. Lee, and T. C. Chung, “Accurate multi-criteria decision making methodology for recommending machine learning algorithm,” *Expert Systems with Applications*, vol. 71, pp. 257–278, 2017.