Research Article

Eigen Solution of Neural Networks and Its Application in Prediction and Analysis of Controller Parameters of Grinding Robot in Complex Environments

Shixi Tang,1,2 Jinan Gu,1 Keming Tang,2 Wei Ding,1 and Zhengyang Shang1

1Mechanical Information Research Center, Jiangsu University, Zhenjiang, Jiangsu 212013, China
2School of Information Engineering, Yancheng Teachers University, Yancheng, Jiangsu 224002, China

Correspondence should be addressed to Jinan Gu; gujinan@tsinghua.org.cn

Received 11 May 2018; Revised 27 August 2018; Accepted 10 September 2018; Published 3 January 2019

Guest Editor: Andy Annamalai

The robot dynamic model is often rarely known due to various uncertainties such as parametric uncertainties or modeling errors existing in complex environments. It is a key problem to find the relationship between the changes of neural network structure and the changes of input and output environments and their mutual influences. Firstly, this paper defined the conceptions of neural network solution, neural network eigen solution, neural network complete solution, and neural network partial solution and the conceptions of input environments, output environments, and macrostructure of neural networks. Secondly, an eigen solution theory of general neural networks was proposed and proven including consistent approximation theorem, eigen solution existence theorem, consistency theorem of complete solution, the partial solution, and none solution theorem of neural networks. Lastly, to verify the eigen solution theory of neural networks, the proposed theory was applied to a novel prediction and analysis model of controller parameters of grinding robot in complex environments with deep neural networks and then build prediction model with deep learning neural networks for controller parameters of grinding robot. The morphological subfeature graph with multimoment was constructed to describe the block surface morphology using rugosity, standard deviation, skewness, and kurtosis. The results of theoretical analysis and experimental test show that the output traits have an optional effect with joint action. When the input features functioning in prediction increase, higher predicted accuracy can be obtained. And when the output traits involving in prediction increase, more output traits can be predicted. The proposed prediction and analysis model with deep neural networks can be used to find and predict the inherent laws of the data. Compared with the traditional prediction model, the proposed model can predict output features simultaneously and is more stable.

1. Introduction

Grinding is one of the most environmental-related manufacturing processes in complex environments. These processes are often scattered, inefficient, contaminated, labor-intensive, and foremost field-intensive. And all the processes account for 12–15% of overall manufacturing costs and 20–30% of total manufacturing time [1]. The surface roughness of the casting block and the grinding force prediction are an important aspect of the grinding process for optimization and monitoring. The grinding process is controlled by the grinding robot servo control system. Neural networks can be helpful in reducing the data dimensionality, and the optimization of neural network training may be employed to enhance the learning and adaptation performance of robots [2]. With its powerful approximation ability, neural network has been utilized in many promising fields, such as modeling and identification of complex nonlinear systems and optimization and automatic control. The components integrated with a complex system may interact with each other and bring difficulties to the control [3]. So many grinding robot servo control systems are constructed with neural networks.

Adaptive neural network controllers were studied in many aspects. The approximation-based controllers are designed for induction motors with input saturation [4]
and 3-DOF robotic manipulator that is subject to backlash like hysteresis and friction [5]. The parameter-based controllers are designed to identify the unknown robot kinematic and dynamic parameters for robot manipulators with finite-time convergence [6] and perform haptic identification for uncertain robot manipulators [7]. The predictive controllers are designed for behaviour predicting of electronic circuits [8], trajectory tracking [9], trajectory tracking of underactuated surface vessels [10], performing attitude tracking control of a quad tilt rotor aircraft [11], and tracking error converging to a neighborhood of the origin [12]. A federal Kalman filter based on neural networks is used in the velocity and attitude matching of transfer alignment [13].

An iterative neural dynamic programming is provided for velocity and attitude matching of transfer alignment [13].

Different controller frameworks of neural networks are constructed for different nonlinear systems. A general framework of the nonlinear recurrent neural network was proposed for solving the online generalized linear matrix equation with global convergence property [15]. Another framework, which combines modified frequency slice wavelet transform and convolutional neural networks, was proposed for automatic atrial fibrillation beat identification [16]. A stability criterion of impulsive stochastic reaction-diffusion cellular neural network framework was derived via fixed-point theory [17]. The subnetwork of a class of coupled fractional-order neural networks consisting of N identical subnetworks have \((r + 1)^n\) locally Mittag-Leffler stable equilibria [18]. Our work was implemented based on these frameworks.

The concept of convolutional neural networks made the framework implementation of multilayer neural networks possible [19]. A pretraining method was proposed for hidden layer node weights using unsupervised restricted Boltzmann machine to solve the overfitting problem in gradient descending of multilayer neural networks [20]. A Dropout strategy was proposed to provide a high-efficiency training method of deep neural network weights with nonsaturating neurons, and this method can prevent overfitting and improve accuracy [21]. A reinforcement learning theory was introduced into deep learning neural networks to improve the reasoning ability of deep learning neural networks [22]. Based on these, the above developments were systematically described and were formally named as deep learning [23]. The improved variants of deep learning theory [24] also appeared. Deep learning theory has been widely used because of its high efficiency and high accuracy [25]. The Go program (AlphaGo) was constructed successfully using the deep learning neural networks and successfully defeated the human Go experts [26]. Depth learning theory has reached satisfactory success in the adaptive output feedback control of unmanned aircraft [27], image quality assessment [28], image character recognition under natural environment [29], change detection of image [30], image classification [31], and robotic guidance [32]. A semiactive nonsmooth control method with deep learning was proposed to suppress harmful effect on building structure by surface motion [33]. So the deep neural networks could be used in controller to process massive amounts of unsupervised data in complex scenarios. Neural networks have been widely studied and applied because of their ability to solve complex linear indivisibility problems, especially their super deep learning ability, which has significant advantages in cross-domain big data analysis with unstructured and unidentified patterns. Further, the conditions of existence and global exponential synchronization of almost periodic solutions of the delayed quaternion-valued neural networks were investigated [34], and the function projective synchronization problem of neural networks with mixed time-varying delays and uncertainties asymmetric coupling was investigated [35]. Neural networks have made great achievements in the determination of the success or failure and improving the performance.

But the robot dynamic model is often rarely known due to the complex robot mechanism, let alone various uncertainties such as parametric uncertainties or modeling errors existing in the robot dynamics. Therefore, it is a key problem to find the relationship between the changes of neural network structure and the changes of input and output environments to find their mutual influence. What are the relationships between the changes of neural network structure and the changes of input and output environments? How do they have mutual influence? These are the fundamental credit assignment problems [36]. It is a hotspot in the current research on the interaction with environments of neural networks, especially in input and output data interaction with neural networks [37]. There exist problems to be solved: (1) the quantitative description of input environments, macrostructure of neural networks, and output environments; (2) the quantitative description of the relationship between input environments, macrostructure of neural networks, and output environments; and (3) experimental verification of the relationship between input environments, macrostructure of neural networks, and output environments. Solution to these problems provides support for the wider application of neural networks.

In order to solve these problems, our work focuses on the eigen solution of neural networks of robot in complex environments, including kinematic model parameter uncertainties, dynamic model parameter uncertainties, and external disturbances by directly mapping the morphological subfeatures with controller parameters of the grinding robot. The concept and the eigen solution theorem of neural networks were studied in the second part of the paper. The third part gave the application of the neural network eigen solution rule in prediction analysis of controller parameters of the grinding robot to verify the theory correctness from a practical point of view. In the last part of the paper, we summarized our research work. The main contributions in this paper include the following:

(1) In order to describe the relationship between the change of neural networks, the change of input environments, and the change of output environments, this paper constructed the neural network eigen solution theory, which defined the conception of

(2) In order to verify the eigen solution theory of neural networks, the morphological subfeature graph with multimoment was constructed to describe the block surface morphology using rugosity \( \{ M_{ij,1} \} \), standard deviation \( \{ M_{ij,2} \} \), skewness \( \{ M_{ij,3} \} \), and kurtosis \( \{ M_{ij,4} \} \) and then build prediction model with deep learning neural networks for controller parameters of the grinding robot. The predicted accuracies of the proposed model are more than 95%. Compared with the traditional prediction model, the proposed model can predict output features simultaneously and is more stable.

(3) The experimental results verify the correctness of the neural network eigen solution theory. The results of theoretical analysis and experimental test showed that the output traits have an optional effect with joint action. When the input features functioning in prediction increase, higher predicted accuracy can be obtained. And when the output traits involve in prediction increase, more output traits can be predicted. The deep learning model can find and predict the inherent laws of the data. Moreover, if the analyzed object data are gained in random, there is no nonrandom feature. The deep learning controller provides random predicted result. The output accuracies of random-output experiments are between 65% and 70% but not 50% due to superb affection.

2. Solution of Neural Networks

In our study, we try to find the essential relationship between the changes of neural network structure and the changes of input and output environments and their mutual influence out of the kinematic model parameter uncertainties, dynamic model parameter uncertainties, and external disturbances. We first defined the neural networks from the input and output environments’ perspective, and consistent approximation theorem is proposed with this concept. Then, the concept of solution of neural networks was defined based on this concept, and the corresponding concepts of isomorphism solution, eigen solution, complete solution, and partial solution were defined too. With these concepts and theorems, the eigen solution existence theorem, the consistency theorem of complete solution and partial solution, and no solution theorem were proposed. The relation of definitions and theorems were described in detail (see Figure 1).

**Definition 1 Neural networks.** The neural networks are defined as \{ **Input, Net, Output** \} (see Figure 2). **Input** = \{ \( S_i \) \} is the input object set. **Net** = \{ \( W_{ij} \), \( b_j \) \} are the network layers \((l > 0, i > 0, j > 0)\), and \( W_{ij} \) is the \( k \)-th convolutional kernel \([i \times j]\) of the \( l \)-th layer. If the network does not use convolution, let \( k = 1 \), then \( i \) is the \( i \)-th element in the \((l - 1)\)-th layer, \( j \) is the \( j \)-th element in the \( l \)-th layer, and \( W_{ij} \) is the weight between the \( i \)-th element and the \( j \)-th element. \( b_j \) is bias in the \( l \)-th layer. **Output** = \{ \( T_i \) \} is the output target vector set. \( \psi \) is the feature acquisition function. \( f_j \) is the neural network function. \( f \) is the target function corresponding to the target vector set \{ \( T_i \) \}.

**Definition 2 Solution of neural networks.** In the study, \{ \( \text{Net}_{\text{tr}} \) \} is a trained neural network, \{ \( \text{Input}_{\text{test}} \) \} is a given test set, and \{ \( \text{Output}_{\text{test}} \) \} is the output target vector set corresponding to...
the test set for the trained neural networks. Let \{\text{Output}_{\text{given}}\}
be the target vector, then Error is given as

\[
\text{Error} = \text{Max} \left\{ e_{ip} = \left| T_{i,\text{test}} - T_{i,\text{given}} \right|_{p}, S_{i,\text{test},p} \right\},
\]

where \( p \) is the given number of test sets. Suppose that the minimum threshold is \( T_{\text{threshold}} \); if the Error is higher than the threshold, then the trained neural networks cannot meet the actual needs.

If \( \text{Error} < T_{\text{threshold}} \), then \{Net\}_{r} is called a solution of neural networks \{Input, Net, Output\}.

If \( \text{Error} = 0 \), \{Net\}_{r} is called an ideal solution or a theoretical solution of neural networks \{Input, Net, Output\}.

If \( 0 < \text{Error} < T_{\text{threshold}} \), \{Net\}_{r} is called an approximate solution of neural networks \{Input, Net, Output\}.

If \( \text{Error} > 0 \), we call that the neural networks \{Input, Net, Output\} have no solution.

Definition 3 Isomorphism solution and eigen solution of neural networks. The neural networks \{W_{l,k,ij}, b_{l}\} are trained independently using \( m \) different sets of input object vector sets \{\tilde{S}_{r}\} to obtain \( m \) different solutions \{Net\}_{r}, \( r = 1, \ldots, m \). All these solutions are called isomorphism solutions of neural networks \{W_{l,k,ij}, b_{l}\} for each other.

Let the input object vector set \{\tilde{S}_{r}\} have the solution \{Net\}_{r}, corresponding to the neural networks \{W_{l,k,ij}, b_{l}\}. \{\tilde{S}_{r}\} is given as

\[
\left\{ \tilde{S}_{r} \right\} = \psi(\{S_{r}\}), \quad r = 1, \ldots,
\]

where \( \psi \) is the eigen acquisition function. The original input object vector set \{S_{r}\} is transformed into the feature data object vector set \{\tilde{S}_{r}\}. If \{\tilde{S}_{r}\} has the solution \{Net\}_{r}, solution \{Net\}_{r}, is called eigen solution of \{Net\}_{r}. The eigen solutions \{Net\}_{r} and \{Net\}_{r} are the isomorphism solutions for each other. In the study, we focus on the neural network eigen solution.

Definition 4 Complete solution and partial solution of neural networks. With the output target vector set \{T_{i}\}, \( i \in [1, \text{Max]} \), if the output target vector set includes all the target vectors \{T_{i}\}_{\text{Max}} , then \{Net\}_{\text{Max}} is called the complete solution of neural networks \{W_{l,k,ij}, b_{l}\}. If the output target vector set is part of the target vector \{T_{i}\}_{\text{Part}}, then \{Net\}_{\text{Part}} is called the partial solution of neural networks \{W_{l,k,ij}, b_{l}\}.

Theorem 1 Consistent approximation theorem. First, let the neural networks contain \( l \) layers; each layer has \( n \) hidden layer nodes; each activation function \( g_{i,j} \) of the hidden layer node is a bounded continuous segmentation function, then the function of the neural networks are \( f_{i} \) as

\[
f_{i} = \sum_{j=1}^{n} b_{ij} g_{ij}.
\]

Let the output objective vector set \{T_{i}\} have the corresponding objective function \( f \). The residual function \( e_{i} \) between the neural networks and the output target vector is defined as

\[
e_{i} = f - f_{i}.
\]

If \( \int_{a}^{b} g(x)dx \neq 0 \), \( f \) and \( f_{i} \) are continuous; the limit of \( e_{i} \) is given as

\[
\lim_{n \to \text{coerT-voi}} || e_{i} || = 0.
\]

That is, \( e_{i} \) converges and monotonically declines.

Proof

(1) If the neural networks are feedforward neural networks and the hidden layer number is \( l = 1 \), then Theorem 1 is proved true with the consistent approximation theorem of feedforward neural networks proposed in [38]

(2) If the neural networks are feedforward neural networks and the hidden layer number is \( l \neq 1 \), then \( f_{i} \) is given as

\[
f_{i} = \sum_{j=1}^{n} \sum_{i} b_{ij} g_{ij} = \sum_{i} b_{ij} g_{ij} = \sum_{i} b_{ij} g_{ij}.
\]

Then, the converted problem is consistent with the consistent approximation theorem of feedforward...
neural networks proposed in [39]; hence, Theorem 1 is true.

(3) If the neural networks are recurrent neural networks \( NET \), the feedback structure of recurrent neural networks is expanded in time dimension. Let the network \( NET \) run from time \( t_0 \); every time step of network \( NET \) is expanded into one of the layers in the feedforward neural networks \( NET^* \), and the layer has exactly the same activity value with time step network \( NET \). For any \( t \geq t_0 \), the connection weight \( w_{ij} \) between the neuron or external input \( j \) and neuron \( i \) of network \( NET \) is copied into \( w_{ij}(t + 1) \) between the neuron or external input \( j \) in the \( t \)-th layer and neuron \( i \) in the \( (t + 1) \)-th layer of network \( NET^* \) (see Figure 3).

As time goes by, the depth of \( NET^* \) can be infinitely deep, that is, the recurrent neural networks \( NET \) can be equivalent to the feedforward neural networks when \( n \to \infty \), and this is exactly the same with the consistent approximation theorem of feedforward neural networks proposed in [38], which also validates Theorem 1.

**Theorem 2 Eigen solution existence theorem.** If the neural networks \( \{W_{lkij}, b_i\} \) have solution \( \{\text{Net}_{t_i}\} \), then for neural networks \( \{W_{lkij}, b_i\} \), there exists the eigen solution \( \{\text{Net}_{t_i}\}_r \).

**Proof.** Take the eigen acquisition function \( \psi(x) = x \), then \( \{S\}_r \) is given as
\[
\{S\}_r = f(\{S\}_l) = \{S\}_l.
\] (7)

So \( \{\text{Net}_{t_i}\}_r \) is given as
\[
\{\text{Net}_{t_i}\}_r = \{\text{Net}_{t_i}\}_r.
\] (8)

Proof finished.

**Theorem 3 Consistency theorem of complete solution and partial solution.** If the neural networks \( \{W_{lkij}, b_i\} \) have solution \( \{\text{Net}_{t_i}\}_r \), then for neural networks \( \{W_{lkij}, b_i\} \), there exists the eigen solution \( \{\text{Net}_{t_i}\}_r \). If the partial solution \( \{\text{Net}_{t_i}\}_\text{Part} \) is gained by training the neural networks \( \{W_{lkij}, b_i\} \) with the input processing object vector set \( \{S\}_i \), then the complete solution \( \{\text{Net}_{t_i}\}_\text{Max} \) can be gained by training the neural networks \( \{W_{lkij}, b_i\} \) with the input processing object vector set \( \{S\}_i \). The consistency theorem of complete solution and partial solution shows that the output traits have an optional effect with joint action. The more the output traits are involved in prediction, the more the output traits can be predicted.

**Proof.** The corresponding target function of the output target vector set \( \{T_{ij}\}_\text{Part} \) is \( f \), and the part solution is \( \{\text{Net}_{t_i}\}_\text{Part} \) corresponding to the mapping function \( f_{ij} \) of neural networks \( \{W_{lkij}, b_i\} \). The solution exists, so the approximation of \( e_l = 0 \) holds; the convergence and monotone decrease of \( e_l \) also hold, \( f_l \) and \( f_{ij} \) are continuous too.

When the output target vector set is \( \{T_{ij}\}_\text{Max} \), the target function is still \( f \), and the complete solution is \( \{\text{Net}_{t_i}\}_\text{Max} \) corresponding to the mapping function \( f_{ij} \) of neural networks \( \{W_{lkij}, b_i\} \), but the target vector set nodes increase (see Figure 4).

Therefore, the number of effective hidden nodes is increased. According to the consistent approximation theorem, \( e_l = 0 \) is more strictly approximated, that is, the trained neural networks \( \{W_{lkij}, b_i\} \) with input object vector set \( \{S\}_i \) can also get a complete solution \( \{\text{Net}_{t_i}\}_\text{Max} \).

Proof finished.

**Theorem 4 No solution theorem.** If the given trained neural networks \( \{W_{lkij}, b_i\} \) with input processing object vector set \( \{S\}_i \) have no complete solution, then the trained neural networks \( \{W_{lkij}, b_i\} \) with input processing object vector set \( \{S\}_i \) have no partial solution. The no solution theorem shows that if the analyzed data objects are random and there is no nonrandom trait, then the predicted result of the deep learning controller is also random; the controller cannot find nonrandom rule with random inputs. In other words, the prediction of deep learning controller cannot be out of nothing.

**Proof.** When the output target vector set is \( \{T_{ij}\}_\text{Max} \), the target function is \( f \), and the complete solution corresponding to the mapping function \( f_{ij} \) of neural networks \( \{W_{lkij}, b_i\} \) has no solution, so \( e_l = 0 \) does not hold.

The target function corresponding to target vector set \( \{T_{ij}\}_\text{Part} \) is \( f \); the input processing object vector set for training neural networks \( \{W_{lkij}, b_i\} \) is \( \{S\}_i \). Here, the number of target vector set nodes is reduced; the number \( n \) of effective nodes in the hidden layer is reduced, so \( e_l = 0 \) does not hold more strictly, i.e., the trained neural networks
{W_{lk,jr}, b_1} with input processing object vector set {S_i} have no partial solution.

Proof finished.

3. Prediction and Analysis of Controller Parameters of Grinding Robot

3.1. Grinding Robot Controller Model. In grinding robot servo control system, dynamic real-time adaptive positioning mechanism was added to the traditional robot servo control system driven by the machine vision system. Our grinding robot controller model describes the closed-loop control process for robot grinding (see Figure 5).

The dynamic real-time adaptive specific controlling method of grinding robot is that the image of block surface is obtained using vision measurement and positioning system in time \( t \). \( L_i \) and \( W_i \) are the length and width of corresponding defective area waiting to handle. The corresponding handling is done to obtain an appropriate knowledge feature value \( S_i \) given as

\[
S_i = I_i (L_i, W_i, F_{\text{flat}}).
\]

In our research, we set \( S_i = L_i^* W_i^* F_{\text{flat}} \); \( F_{\text{flat}} \) is the roughness of casting block surface.

The speed of robotic arm end coincides with the speed of grinding wheel, \( V_c(t) = v_p(t) = F_1(S_i) = k_1S_i \), \( V_w(t) = F_2(S_i) = k_2S_i \), \( f_a(t) = F_3(S_i) = k_3S_i \), \( f_r(t) = F_4(S_i) = k_4S_i \); the coefficients \( k_1 \)–\( k_4 \) are determined by experiment based on casting block material.

The robot controller achieves adaptive control by the dynamic real-time feedback information wherein \( F_r \) is the reference force, \( q_r \) and \( q_r - n \) are the joint angles, \( q'_r \) and \( q'_r - n \) are the robot joint angular velocities, \( \tau \) is the joint driving moment, \( f \) is the force applied to the casting block by the end of the robot, \( q_m \) is the feedback joint angle, and \( q'_m \) is the feedback joint angular velocity.

The working process algorithm of the grinding robot controller model describes the closed-loop control process (see Figure 6).

First, the surface image \( I \) of block is gained with vision measurement and positioning system, and then, the geometrical features of image \( I \) is obtained with vision measurement and positioning system too. By judging the quality of the casting block, the grinding decision is made. When the surface of block does not meet the quality requirement, the robot grinding traits are got from the geometrical features of image \( I \) with robot grinding controller in the grinding robot servo control system. Last, the grinding robot executes the grinding operation with the robot grinding traits, and the surface image \( I \) of block feeds back to the vision measurement and positioning system to form a closed loop.

The speed of robotic arm end coincides with the speed of grinding wheel, \( V_c(t) = v_p(t) = F_1(S_i) = k_1S_i \), \( V_w(t) = F_2(S_i) = k_2S_i \), \( f_a(t) = F_3(S_i) = k_3S_i \), \( f_r(t) = F_4(S_i) = k_4S_i \); the coefficients \( k_1 \)–\( k_4 \) are determined by experiment based on casting block material.

The robot controller achieves adaptive control by the dynamic real-time feedback information wherein \( F_r \) is the reference force, \( q_r \) and \( q_r - n \) are the joint angles, \( q'_r \) and \( q'_r - n \) are the robot joint angular velocities, \( \tau \) is the joint driving moment, \( f \) is the force applied to the casting block by the end of the robot, \( q_m \) is the feedback joint angle, and \( q'_m \) is the feedback joint angular velocity.

The working process algorithm of the grinding robot controller model describes the closed-loop control process (see Figure 6).

First, the surface image \( I \) of block is gained with vision measurement and positioning system, and then, the geometrical features of image \( I \) is obtained with vision measurement and positioning system too. By juding the quality of the casting block, the grinding decision is made. When the surface of block does not meet the quality requirement, the robot grinding traits are got from the geometrical features of image \( I \) with robot grinding controller in the grinding robot servo control system. Last, the grinding robot executes the grinding operation with the robot grinding traits, and the surface image \( I \) of block feeds back to the vision measurement and positioning system to form a closed loop.

The working process algorithm describes the specific controlling method of grinding robot controller (see Figure 7).

After the geometrical features of image \( I \) are obtained, robot grinding traits are generated with grinding robot servo control system in the controller, which is composed of...
prediction model of trained deep neural networks. Then, the joint angles are computed from robot grinding traits with kinematics module, and the robot joint angular velocities are computed from the joint angles with force control module. The grinding robot executes the grinding operation with the robot grinding traits, joint driving moment, and joint angular velocities.

3.2. Prediction Model of Controller Parameters of Grinding Robot. The research object in our study is the robot grinding process of casting block of engine. Block is one of the important parts of the engine. The quality of casting block engine directly affects the automobile performance. With the development of automobile engine technology, the dimensional accuracy and mechanical performance of engine block are required for high-quality casting block. The automobile engine of study is 1.5 L; maximum outline dimension of the block is 400 mm × 320 mm × 253 mm, and its weight is 38 kg of HT300 material.

The controller parameters of grinding robot affect grinding efficiency and quality which includes the feed speed \( V_c \) of grinding wheel translational motion, the peripheral speed \( V_w \) of grinding wheel rotational motion, the axial displacement \( f_a \) of workpiece motion of the block, and the
radial displacement $f_r$ of wheel motion (see Figure 8). The specific controller parameters of grinding robot are the output data, and their type and number are related to the machining method. The block is located on the bracket of track, while the grinding wheel is located on the actuator side of the robot. During the process, the workpiece is fixed, whereas the grinding wheel moves.

According to DIN EN ISO 4287 and ASME B46.1 standards, the input parameters are $\text{Fi} = (R_a, R_z, R_y)$ features of workpiece surface morphology; $R_a$ is the arithmetical mean deviation of the workpiece surface morphology; $R_z$ is the point height of irregularities of the workpiece surface morphology, and $R_y$ is the maximum height of the workpiece surface morphology. The value of $R_a$ of the workpiece surface after grinding ranges from 0.01 $\mu$m to 0.8 $\mu$m [40]. Features ($R_a, R_z, R_y$) are the statistical extract and dimensionality of surface morphology and cannot characterize all the surface morphology. Features ($R_a, R_z, R_y$) are only the empirical description of surface morphology and cannot characterize essential features of surface morphology. Features ($R_a, R_z, R_y$) are only the statistical average of surface morphology with a lot of information loss. The standards of DIN EN ISO 4287 and ASME B46.1 give more explicit and more information of surface morphology using four features, which are surface rugosity, standard deviation, skewness, and kurtosis [39, 41, 42], presented in (10)–(13). But these features still cannot work on all the pixels of surface morphology and cannot give a significant and systematic description statistics.

$$R_a = \frac{1}{I} \int_0^1 |Z(x)|dx,$$  \hspace{1cm} (10)

$$R_q = \sqrt{\frac{1}{I} \int_0^1 Z^2(x)dx},$$  \hspace{1cm} (11)

$$R_{sk} = \frac{1}{R_q^2} \int_0^1 Z^3(x)dx,$$  \hspace{1cm} (12)

$$R_{k4} = \frac{1}{R_q^4} \int_0^1 Z^4(x)dx.$$  \hspace{1cm} (13)

In order to solve the above problems, we improved the definition of surface rugosity, standard deviation, skewness, and kurtosis in [26–28]. Alternatively, we give a unified definition of surface morphological features with definite mathematical meanings; the features are corresponding to the first moment, second moment, third moment, and fourth moment, respectively. On the other hand, the definitions of features are extended to all pixels of surface morphological image, which can reduce the information loss to optional extent, while extracting the essential features at the same time. The features are defined in (14)–(17) by improving the method of calculating the depth from the gray level [43].

$$I_{ij} = \sqrt{\frac{\gamma \times \ln \times \cos \alpha}{g_{ij}}} - f - u_A,$$  \hspace{1cm} (14)

$$\mu = \frac{1}{A \times B} \sum_{i=1}^{A} \sum_{j=1}^{B} I_{ij},$$  \hspace{1cm} (15)

$$\sigma = \sqrt{\frac{1}{A \times B} \sum_{i=1}^{A} \sum_{j=1}^{B} (I_{ij} - \mu)^2},$$  \hspace{1cm} (16)

$$M_{ij} = \frac{(I_{ij} - \mu)^k}{\sigma^k}, \quad k = 1, 2, 3, 4.$$  \hspace{1cm} (17)

Here, $\gamma$ is the reflectivity of the block surface to the incident light. $\times$ is the intensity of block surface. In $\times \cos \alpha$ is the vertical component of the intensity of block surface. $f$ is the focal length of the camera. $u_A$ is the object distance from camera to block surface. $g_{ij}$ is the gray value of pixel of block surface image $I$ in position $(i, j)$. $I_{ij}$ is the grey value of pixel of block surface image $I$ in position $(i, j)$. $A$ and $B$ are the maximum values of coordinates $i$ and $j$; $\mu$ is the average depth of all pixels of block image, and $\sigma$ is the mean square deviation of the depth of all pixels of block image. $M_{ij,k}$ is the moment of the depth of the pixel in position $(i, j)$, $M_{ij,1}$ is the first moment of the depth of the pixel in position $(i, j)$, denoting the rugosity of surface image in position $(i, j)$. $M_{ij,2}$ is the second moment of the depth of the pixel in position $(i, j)$, representing the standard deviation of surface image in position $(i, j)$. $M_{ij,3}$ is the third moment of the depth of the pixel in position $(i, j)$, signifying the skewness of surface image in position $(i, j)$. $M_{ij,4}$ is the fourth moment of the depth of the pixel in position $(i, j)$, indicating the kurtosis of surface image in position $(i, j)$.

3.3. Grinding Robot Prediction Algorithm of Controller Parameters of Grinding Robot. Grinding parameter prediction is completed by the trained deep neural networks (see Figure 9). The trained deep neural networks are part of the grinding robot controller. The surface image $I$ of part of block is analyzed by algorithm, then the geometrical features of the obtained surface image are obtained, including the
rugosity $M_{ij,1}$, standard deviation $M_{ij,2}$, skewness $M_{ij,3}$, and kurtosis $M_{ij,4}$ of surface image $I$ in position $(i,j)$. The feature values are put into the well-trained deep neural networks, and the values of robot grinding traits are obtained when the network computing is finished. Then, the grinding robot controller sends out the control instruction to other institutions of the robot.

In the constructed model, the surface roughness of block surface was obtained by analyzing the surface of the workpiece, including the corresponding moment features. All the moment features of image are input into the grinding controller, which determines the feed speed $V_c$, the peripheral speed $V_w$, the radial displacement $f_a$, and the radial displacement $f_r$ of the grinding robot actuator.

It is a complex nonlinear relationship between rugosity $M_{ij,1}$, standard deviation $M_{ij,2}$, skewness $M_{ij,3}$, kurtosis $M_{ij,4}$ and feed speed $V_c$, peripheral speed $V_w$, axial displacement $f_a$, and radial displacement $f_r$ in (18). The nonlinear relationship between them is described using deep neural networks.

$$F_{out}(V_c, V_w, f_a, f_r) = f(M_{1}, M_{2}, M_{3}, M_{4}). \quad (18)$$

Further, we give the process of generating the deep neural networks (see Figure 10). The feature subgraphs $M_{ijk}$ of part of block surface image $I$ is generated after the surface image $I$ being processed with the grinded part of block. The initial values of weights and biases are initialized in deep neural networks. The deep neural networks are trained using the $(I, F_{outs})$ training set that had been marked with empirical knowledge. $I$ is the block images and $F_{outs}$ are the output of the feature subgraphs corresponding to $I$. Hence, the trained deep neural networks are fine-tuned to get one stage result of deep neural networks. This iterating process continues until all the data in training set are completed to get the well-trained deep neural networks.

Our proposed model has the characteristics of multifeatures and multitargets, which makes the model complicated. It needs to consider the correlation between rugosity, standard deviation, skewness, kurtosis and their influence on deep neural networks. It also needs to consider the correlation between feed speed $V_c$, peripheral speed $V_w$, axial displacement $f_a$, radial displacement $f_r$, and their influence on deep neural networks. Further, the correlation between rugosity, standard deviation, skewness, kurtosis and feed speed $V_c$, peripheral speed $V_w$, axial displacement $f_a$, radial displacement $f_r$ needs to be considered. The agreement of deep neural networks should be considered, that is, the neural network nodes are independent of each other and overfitting phenomena that do not conform to the agreement.

The inputs of our model have several feature subgraphs rather than a single graph. Our model predicts the grinding traits, not just the classification of a given object. Thus, the model processes several feature subgraphs to predict several grinding traits with multi-input and multi-output relationship, which is different from one to one relationship of single predetermined object classification.

3.4. Grinding Robot Prediction Algorithm of Controller Parameters of Grinding Robot. The images of block surface after grinding are taken using vision system, and all the images are labeled with corresponding grinding trait parameter values, as shown in (17). There are 400 labeled training data measured by the experiment according to years of working experience as skilled technicians.

In (14), $g_{ij}$ is the gray value of block surface image; the reflectance $y$ of block surface is 0.65; the focus $f$ is 50 mm; the object distance to block surface is 0.5 m; $\alpha$ takes 0 for the use of uniform light source, and the intensity of block surface In is 1000 lx. The extended 3D shape of block surface and its feature are obtained by (17). $\text{Gray, } Ra, \overline{M}_1, \overline{M}_2, \overline{M}_3$, and $\overline{M}_4$ are gray level, roughness, rugosity, standard deviation, skewness, and kurtosis; all these features are the inputs of deep neural network controller.

We obtain the correlation between the input feature parameters of the neural network controller (see Figure 11). It can be seen that the correlations among the parameters are very complex, with features of positive and negative correlations, and the degrees of correlation are also different and cannot be expressed with analytical formula.

And the correlation between the output parameters of the neural network controller is also obtained (see Figure 12). The output parameters act as robot grinding traits. It can be seen that the correlations among the parameters are also
Figure 10: The process of generating the deep neural networks.

Figure 11: The correlation between the input feature parameters.
very complicated with features of both positive and negative correlations. The degrees of correlation are also different and cannot be expressed with analytical formula.

3.5. Predicted Result Analysis of Controller Parameters of Grinding Robot. In the experiment, the training time of deep learning neural networks for block grinding is finished within 78 hours and 32 minutes with computing environment of Windows Server 2010, Intel Core i5-4308U CPU, 2.80 GHz, and RAM 8.00 GB. The prediction time of trained deep learning neural networks for block grinding is computed within 2 seconds.

In our controller, the model is implemented on deep learning open-source framework Tiny-dnn [44]. In deep learning neuron network prediction training model, the minibatch_size is 4, the num_epochs are 3000, the activation function is tanh, and its value ranges from $-0.8$ to $+0.8$. In the input layer, the input labeled data are 400, and the default normalized values $[n_{\text{length}} \times n_{\text{width}}]$ are $[32 \times 32]$. In the feature extraction layer, convolution and sampling are performed for each feature subgraph; the convolution kernel is $[5 \times 5]$, and the sampling kernel takes $[2 \times 2]$. In the fuzzy analysis layer, the number of jump positions $n_s$ is 10, the quantum interval initial value $\theta_j$ is 0.1, and the output layer is with full join. The inputs include subgraphs of rugosity $\{M_{ij,1}\}$, standard deviation $\{M_{ij,2}\}$, skewness $\{M_{ij,3}\}$, and kurtosis $\{M_{ij,4}\}$ of the block surface morphology, and each feature subgraph works independently. The full join between the fuzzy quantum analysis layer and output layer of deep neuron network is processed with all the feature subgraphs. The outputs are the feed speed $V_c$, the peripheral speed $V_w$, the axial displacement $f_a$, and the radial displacement $f_r$.

3.5.1. Deep Learning Neuron Network Training Results. The training results of deep learning neuron network of block surface controller parameters of grinding robot are obtained with samples increasing (see Figure 13).

The accuracy of all the output features is more than 80% with the prediction of deep learning neuron network controller under the given empirical labeled data set. The increasing rate of grinding trait accuracy is among the fastest stage when the samples are less than 600. The accuracy of all the grinding traits reaches more than 95% when samples are 1000, and the accuracy still slowly increases with the increasing of training times. In the final training result, the prediction accuracies of feed speed $V_c$, peripheral speed $V_w$, axial displacement $f_a$, and radial displacement $f_r$ are above 98% compared to the labeled training data set.

Similarly, the training results of deep learning neuron network of block surface controller parameters of grinding robot are obtained with training times increasing (see Figure 14).

The accuracy of all the output features is more than 80% with the prediction of deep learning neuron network controller under the given empirical labeled data set. The increasing rate of grinding trait accuracy is among the fastest stage when training times are less than 600. The accuracy of all the grinding traits reaches more than 95% when training times are 1500, and the accuracy still slowly increases with the increasing of training times. In the final training result, the prediction accuracies of feed speed $V_c$, peripheral speed $V_w$, axial displacement $f_a$, and radial displacement $f_r$ are above 97% compared to the labeled training data set.

From the training results of neural networks in Figures 13 and 14, it can be seen that the prediction accuracy of the deep
learning neural networks increases monotonically with the increase of sample size and number of training. And the sample size displays a bigger influence on the accuracy of simulating result than the number of training.

The prediction error of feed speed $V_c$, peripheral speed $V_w$, axial displacement $f_a$, and radial displacement $f_r$ with the deep learning neural networks can be defined as

$$e_{V_c} = 1 - \text{accurate}_{V_c}, \quad (19)$$
$$e_{V_w} = 1 - \text{accurate}_{V_w}, \quad (20)$$
$$e_{f_a} = 1 - \text{accurate}_{f_a}, \quad (21)$$
$$e_{f_r} = 1 - \text{accurate}_{f_r}. \quad (22)$$

From (19)–(22), we can see that $e_{V_c}$, $e_{V_w}$, $e_{f_a}$, and $e_{f_r}$ converge and monotonically decrease. The experimental results verify the correctness of Theorem 1 (consistent approximation theorem).

3.5.2. Predicted Experimental Results with Multioutput. The trained deep neural networks are used to predict the traits of casting block grinding. The sample number of data is 100, which is used as the input of the trained deep neural networks. The predicted results of feed speed $V_c$, peripheral speed $V_w$, axial displacement $f_a$, and radial displacement $f_r$ of grinding wheel are obtained accordingly (see Figures 15–18). The black curve represents the labeled grinding trait data of block surface. The other color curves are the predicted results using the trained deep neural networks, and the average accuracies of predicted results are given (see Table 1).

From Figures 15–18, we can get that $\{g_{ij}\}_{32 \times 32}$, $\{g_{ij}\}_{48 \times 48}$, and $\{g_{ij}\}_{52 \times 52}$ are input data sets of different sizes; $\{M_{ij,1}\}_{32 \times 32}$, $\{M_{ij,2}\}_{32 \times 32}$, $\{M_{ij,3}\}_{32 \times 32}$, and $\{M_{ij,4}\}_{32 \times 32}$ are the feature data set that $\{g_{ij}\}_{32 \times 32}$ transformed, and each input data set corresponds to the respective output data sets $\{V_c, V_w, f_a, f_r\}$. In the given set of input feature data, at least one data set can be found to get the eigen solution $\{Net\}_r$ of solution $\{Net\}_r$ of
Figure 15: Predicted results of feed speed of grinding wheel.

Figure 16: Predicted results of peripheral speed of grinding wheel.

Figure 17: Predicted results of axial displacement of grinding wheel.
neural networks \( \{W_{ijk}, b_i\} \) corresponding to the data set of \( \{g_{ij}\}_{32 \times 32} \) and the prediction accuracy of eigen solution \( \{\text{Net}_{tr}\}_{r} \) is equivalent to that of neural network solution \( \{\text{Net}_{tr}\}_{r} \) corresponding to the data set of \( \{g_{ij}\}_{32 \times 32} \), so the correctness of Theorem 2 (eigen solution existence theorem) is verified. The following conclusions are drawn from Figures 15–18 and Table 1 with the average accuracies of predicted results.

(1) The trained deep learning neural networks of our model can predict the feed speed \( V_c \), peripheral speed \( V_w \), axial displacement \( f_a \), and radial displacement \( f_r \) simultaneously, and their average minimum prediction accuracies are more than 95%. The more pixels the input data samples from the image, the higher the correlation prediction accuracy. The green, blue, and red curves are the correlation predicted results of our model with pixels \( 32 \times 32, 48 \times 48, \) and \( 52 \times 52 \) where the accuracy is improved by 1% in turn

(2) Our proposed deep neuron network can well reflect the influence between the features of rugosity \( M_{ij,1} \), standard deviation \( M_{ij,2} \), skewness \( M_{ij,3} \), kurtosis \( M_{ij,4} \) and the robot grinding traits described as follows. (a) There are several features, at least one feature of rugosity \( M_{ij,1} \), standard deviation \( M_{ij,2} \), skewness \( M_{ij,3} \), kurtosis \( M_{ij,4} \) playing function on all the grinding traits of feed speed \( V_c \), peripheral speed \( V_w \), axial displacement \( f_a \), and radial displacement \( f_r \). (b) The joint prediction accuracy of robot grinding traits with all block surface features is within the same precision level of the optimal prediction accuracy of each independent robot grinding trait as rugosity \( M_{ij,1} \), standard deviation \( M_{ij,2} \), skewness \( M_{ij,3} \), kurtosis \( M_{ij,4} \). Furthermore, the joint prediction accuracy is even sometimes superior to the single feature optimal prediction precision. As seen in Figure 17, the joint prediction accuracy of axial speed is superior to the optimal prediction precision of rugosity \( M_{ij,1} \). All the joint predicted results of robot grinding traits obtain the optimal prediction accuracy at the same time where the joint predicted results do not interfere with each other. (c) Different

![Figure 18: Predicted results of radial displacement of grinding wheel.](image)

**Table 1: Average accuracies of predicted results with multioutput.**

<table>
<thead>
<tr>
<th>Average accuracy</th>
<th>( V_c ) (mm/s)</th>
<th>( V_w ) (m/s)</th>
<th>( f_a ) (( \mu m ))</th>
<th>( f_r ) (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_{ij} )<em>{32 \times 32} ( (A</em>{32}) )</td>
<td>0.967164</td>
<td>0.975184</td>
<td>0.960752</td>
<td>0.950227</td>
</tr>
<tr>
<td>( g_{ij} )<em>{48 \times 48} ( (A</em>{48}) )</td>
<td>0.970117</td>
<td>0.979107</td>
<td>0.958325</td>
<td>0.953480</td>
</tr>
<tr>
<td>( g_{ij} )<em>{52 \times 52} ( (A</em>{52}) )</td>
<td>0.972249</td>
<td>0.980596</td>
<td>0.962713</td>
<td>0.958631</td>
</tr>
<tr>
<td>( M_{ij,1} )<em>{32 \times 32} ( (A</em>{32}) )</td>
<td>0.967741</td>
<td>0.976567</td>
<td>0.955851</td>
<td>0.951604</td>
</tr>
<tr>
<td>( M_{ij,2} )<em>{32 \times 32} ( (A</em>{32}) )</td>
<td>0.944570</td>
<td>0.948410</td>
<td>0.914060</td>
<td>0.907100</td>
</tr>
<tr>
<td>( M_{ij,3} )<em>{32 \times 32} ( (A</em>{32}) )</td>
<td>0.928147</td>
<td>0.929163</td>
<td>0.887634</td>
<td>0.887930</td>
</tr>
<tr>
<td>( M_{ij,4} )<em>{32 \times 32} ( (A</em>{32}) )</td>
<td>0.829836</td>
<td>0.759111</td>
<td>0.745841</td>
<td>0.744105</td>
</tr>
</tbody>
</table>
surface feature has different influence on the robot controller parameters of the grinding robot. Here, the influence of rugosity \( M_{ij1} \), standard deviation \( M_{ij2} \), skewness \( M_{ij3} \), and kurtosis \( M_{ij4} \) are in descending order. As shown in Figures 15, 16, and 18, the average prediction accuracy of rugosity \( M_{ij1} \) is higher than that of the joint prediction accuracy when the dotted curve is the image with pixels \([32 \times 32]\). The predicted results of standard deviation \( M_{ij2} \) and skewness \( M_{ij3} \) have larger amplitude fluctuation, and their accuracies are also less than that of the joint accuracy. The predicted result of kurtosis \( M_{ij4} \) is a line; the corresponding controller almost has no prediction capability, and the prediction accuracy is only a random probability.

3. There are severe deviations in times 78–83 and 87–95 as shown in Figure 15. Also, similar deviation is formed in times 4 and 85–95 as shown in Figure 10. In times 68–72 and 86–92 as shown in Figure 17, as well as in times 66–68 and 84–94 as indicated in Figure 18, severe abnormality were recorded, which are caused by the emergence of new data samples. The solution to this problem is that new data samples are labeled and added to the training set. This model can be used to obtain reasonable predicted result when the similar problem occurs in the future.

Therefore, given the standard image of grinded block surface, we can accurately get the optimal configuration values of the feed speed \( V_c \), peripheral speed \( V_w \), axial displacement \( f_a \), and radial displacement \( f_r \) using the well-trained deep neuron network controller.

3.5.3. Predicted Experimental Results with Single Output. In the independent experiment with single output, the multi-to-one deep learning-oriented prediction model is adopted, and the inputs are the moment feature graphs of grinding casting block surface corresponding to the first, second, third, and fourth moments. Each of the moment feature graphs works in a relatively independent network. The output parameter between the fuzzy analysis layer and the output layer of network is single output with peripheral speed \( V_w \), feed speed \( V_c \), axial displacement \( f_a \), or radial displacement \( f_r \). The predicted results are obtained (see Figures 19–22). The average accuracies of predicted results are given (see Table 2).

In the independent experiment, \( \{g_{ij}\}_{[32 \times 32]} \), \( \{g_{ij}\}_{[48 \times 48]} \), and \( \{g_{ij}\}_{[52 \times 52]} \) are input data sets of different sizes; \( M_{ij1} \), \( M_{ij2} \), \( M_{ij3} \), and \( M_{ij4} \) are the feature data set that \( g_{ij} \) is transformed to, and each input data set corresponds to the respective output data set \( \{V_c\}, \{V_w\}, \{f_a\}, \text{or} \{f_r\} \). The solution of independent single-output experiment is partial solution \( \{\text{Net}_{\text{ij}}\}_{\text{Part}} \) of multi-output experiment complete solution \( \{\text{Net}_{\text{ij}}\}_{\text{Max}} \). The partial solution \( \{\text{Net}_{\text{ij}}\}_{\text{Part}} \) predicted results are shown in Figures 19–22. Data sets \( \{g_{ij}\}_{[32 \times 32]} \), \( \{g_{ij}\}_{[48 \times 48]} \), and \( \{g_{ij}\}_{[52 \times 52]} \) have partial solution for all output data set of \( \{V_c\}, \{V_w\}, \{f_a\}, \text{or} \{f_r\} \). Similarly, \( M_{ij3} \) has partial solution for \( V_w \). In the multi-output experiment, the feature data sets \( \{M_{ij1}\}, \{M_{ij2}\}, \{M_{ij3}\}, \text{and} \{M_{ij4}\} \) of input data sets \( \{g_{ij}\}_{[32 \times 32]} \), \( \{g_{ij}\}_{[48 \times 48]} \), and \( \{g_{ij}\}_{[52 \times 52]} \) have independent multi-output experiment complete solutions \( \{\text{Net}_{\text{ij}}\}_{\text{Max}} \) corresponding to \( \{V_c, V_w, f_a, f_r\} \), as shown in Figures 15–18. All \( \{M_{ij1}\}, \{M_{ij2}\}, \text{and} \{M_{ij3}\} \) have complete solutions \( \{\text{Net}_{\text{ij}}\}_{\text{Max}} \) except for \( M_{ij4} \). From the above, we can draw the conclusion that a partial solution must correspond to a complete solution. Thus, our experiment verifies the correctness of Theorem 3 (consistency theorem of complete solution and partial solution). The independent experiment shows that \( \{M_{ij4}\} \) has no complete solutions \( \{\text{Net}_{\text{ij}}\}_{\text{Max}} \) corresponding to \( \{V_c, V_w, f_a, f_r\} \), thus verifying the correctness of Theorem 4 (no solution theorem).
As shown in Figure 19, the black curve is the original empirical data of the peripheral speed of grinding wheel. The other curves are the predicted results of deep learning-oriented controllers. From the figures and the average accuracies in Table 2, we draw the following conclusions. (1) When output parameter is only the feed speed, the single-output prediction accuracies of each moment are in the same percentage point scope with the multioutput prediction accuracies of each moment, as the blue and the red curve shows; the multioutput prediction accuracies do not interfere with each other for the sake of interaction of multi-output parameters. (2) In this experiment, only the first moment has function, and other moments have no function, as shown with dotted straight lines in the figures. The predicted results of the second, the third, and the fourth moments are only random results. The multioutput prediction accuracies are consistent with the first moment predicted result, which means that the first moment has superb effect in the multioutput prediction. (3) There are 3 nonfunctional moments in single-output experiment, which are 2 more nonfunctional moments than in multioutput experiment. This phenomenon shows that there exists superb effect moment between the moments functioning in multioutput experiment.

The predicted result of feed speed of grinding casting block surface is similar to the predicted result of peripheral speed as shown in Figure 20. The differences are shown as follows. (1) Compared with Figure 19, the second and the fourth moments do not work in this experiment; the predicted experimental accuracy of the fourth moment with
single output is reduced by 10%, which also shows that there exists superb effect moment between the moments functioning in multioutput experiment. (2) The first and third moments function in this experiment, which means that the inputting feature parameters are determined by the grinding surface morph itself but not the order of moments.

From the above analysis, it can be seen that (1) there is existence of high prediction accuracy consistent with multi-input and multioutput prediction, multi-input and single-output prediction, and single-input and single-output prediction. The optimal prediction accuracy is obtained at the same time for all the predicted results, which do not interfere with each other for the superb affection existing. (2) Also, the optimal prediction accuracy consistent with multi-input and multioutput prediction, multi-input and single-output prediction, and single-input and single-output prediction fully demonstrate that the grinding data do have inherent law and can be captured and predicted by the designed deep learning neural network controller. In order to further verify this, we conducted experiment with random input as follows.

3.5.4. Predicted Experimental Results with Random Output.

In the random-output experiment, we adopt the same proposed deep learning-oriented prediction model, and the inputs are the moment graph of the grinding surface morph data. Each moment graph works in relatively independent network. All the output data values of the feed speed \( V_c \), peripheral speed \( V_w \), axial displacement \( f_a \), and radial displacement \( f_r \) are randomly generated by computer simulation. The predicted results of random-output experiment are obtained (see Figures 23–26). The black curves are generated by random data, and the blue curves are the predicted results with our controller. The average accuracies of predicted results with random output are given (see Table 3), and the prediction results are in accordance with the random data. That is to say, we get the random results.

From the analysis of random-output experiment, we get a very interesting result. If the analyzed object data are gained in random, there is no nonrandom feature. The deep learning prediction controller gets random predicted result, which cannot be found in nonrandom laws. In other words, the deep learning prediction controller cannot be out of nothing, which proves the validity of the proposed model from another side. The stronger the law, the more the input features function, the higher the predicted accuracy. The random-output experiment further verifies the correctness of Theorem 4.

![Machining parameter: axial displacement](image)

**Figure 22:** Predicted results of axial displacement of grinding wheel.

<table>
<thead>
<tr>
<th>Average accuracy</th>
<th>( V_c ) (mm/s)</th>
<th>( V_w ) (m/s)</th>
<th>( f_a ) (μm)</th>
<th>( f_r ) (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( {g_0}<em>{[32 \times 32]} ) ( \bar{A}</em>{32} )</td>
<td>0.975184</td>
<td>0.967164</td>
<td>0.950027</td>
<td>0.960752</td>
</tr>
<tr>
<td>( {g_0}<em>{[32 \times 32]} ) ( \bar{A}</em>{32} )</td>
<td>0.977520</td>
<td>0.972553</td>
<td>0.958349</td>
<td>0.964409</td>
</tr>
<tr>
<td>( {M_{i,1}}<em>{[32 \times 32]} ) ( \bar{A}</em>{3iM_i} )</td>
<td>0.979840</td>
<td>0.907778</td>
<td>0.956100</td>
<td>0.938146</td>
</tr>
<tr>
<td>( {M_{i,2}}<em>{[32 \times 32]} ) ( \bar{A}</em>{3iM_i} )</td>
<td>−0.57020</td>
<td>0.763325</td>
<td>0.595728</td>
<td>0.603273</td>
</tr>
<tr>
<td>( {M_{i,3}}<em>{[32 \times 32]} ) ( \bar{A}</em>{3iM_i} )</td>
<td>−0.570200</td>
<td>0.929030</td>
<td>−4.42592</td>
<td>−4.63600</td>
</tr>
<tr>
<td>( {M_{i,4}}<em>{[32 \times 32]} ) ( \bar{A}</em>{3iM_i} )</td>
<td>0.749110</td>
<td>0.763325</td>
<td>0.595728</td>
<td>0.603237</td>
</tr>
</tbody>
</table>
Machining parameter: peripheral speed

Original data
Predicted result with all 32

Figure 23: Predicted results of peripheral speed of grinding wheel.

Machining parameter: feed speed

Original data
Predicted result with all 32

Figure 24: Predicted results of feed speed of grinding wheel.

Machining parameter: radial displacement

Original data
Predicted result with all 32

Figure 25: Predicted results of radial displacement of grinding wheel.
4. Conclusions
In our study, we give the eigen solution theory of general neural networks and quantitatively describe the input environments, the macroscopic structure, output environments, and their relationship of neural networks. The eigen solution theory is applied and validated in the prediction of controller parameters of grinding robot in complex environments with the proposed deep learning neural networks, which will provide theoretical support for a wider application with the proposed deep learning neural networks.

Data Availability
The data used to support the findings of this study are available from the supplementary materials.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
The work was supported by the National Science Foundation of China [51875266, 61379064, 61603326, 61772448], the Industry-Academia Prospect Research Foundation of Jiangsu [BY2016066-06, BY2015058-01], the National Students’ Innovation and Entrepreneurship Training Program of Jiangsu [201310324006, 2103242013006], the Tendered Special Fund Project on Promoting the Transformation of Scientific and Technological Achievements of Jiangsu [BA2015026], the Special Fund Project on Promoting the Transformation of Scientific and Technological Achievements of Jiangsu [KA2015027], and the Research Innovation Program for College Graduates of Jiangsu [KYLX16_0880].

Supplementary Materials
The data sets provided are the images of block surface after grinding taken using vision system. There are 400 train images in the imagetrain set and 100 train images in the imagetest set of inputdata sets. And there are 400 items in the train set and 100 items in the test set of outputdata sets. Each item contains the values of feed speed, peripheral speed, axial displacement, and radial displacement. (Supplementary Materials)

References


[34] Y. Li, X. Meng, and Y. Ye, “Almost periodic synchroniza-
tion for quaternion-valued neural networks with time-var-

[35] T. Botmart, N. Yotha, P. Niamsup, and W. Weera, “Hybrid adaptive pinning control for function projective synchro-


[38] G. B. Huang, L. Chen, and C. K. Siew, “Universal approxima-


