

Research Article

An Intelligent Calculation Method of Volterra Time-Domain Kernel Based on Time-Delay Artificial Neural Network

Kaiyang Zhong ¹ and Lerui Chen ²

¹School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 611130, China

²State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Correspondence should be addressed to Lerui Chen; chenlerui5566@163.com

Received 29 February 2020; Revised 8 April 2020; Accepted 20 April 2020; Published 11 May 2020

Academic Editor: Asier Ibeas

Copyright © 2020 Kaiyang Zhong and Lerui Chen. 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.

To solve the problems of high complexity and low accuracy in Volterra time-domain kernel calculation of a nonlinear system, this paper proposes an intelligent calculation method of Volterra time-domain kernel by time-delay artificial neural networks (TDANNs) and also designs a root mean square error (RMSE) index to choose the neuron number of the network input layer. Firstly, a three-layer TDANN is designed according to the characteristics of the Volterra model. Secondly, the relationship between parameters of TDANN and Volterra time-domain kernel is analyzed, and then three-order expressions of Volterra time-domain kernel are derived. The calculation of Volterra time-domain kernel is completed by network training. Finally, it is verified by a nonlinear system. Simulation results indicate that compared with traditional methods, the new method has higher accuracy, and it can realize the batch calculation of Volterra kernel, which not only improves the calculation efficiency but also provides accurate data for fault diagnosis based on Volterra kernel in further research work.

1. Introduction

As the transfer function of a linear system, Volterra kernel (including Volterra kernel time-domain kernel and Volterra kernel frequency-domain kernel) does not depend on the input and output of the system, which can characterize the essential properties of the nonlinear system and explain the special phenomena of the nonlinear system. It has advantages of clear physical meaning and rich information, which is widely used in electronic engineering, mechanical and electrical engineering, control engineering, and other fields [1–6].

At present, the research and application of Volterra kernel have developed rapidly; especially the nonlinear spectrum research based on the generalized frequency response function (GFRF) and the output spectrum response function (OFRF) has been gradually applied to the fields of feature extraction, mechanism analysis, and fault diagnosis [7–11]; however, the complexity and accuracy of Volterra kernel calculation are the key factors to its application. Until

now, the calculation of Volterra kernel is still difficult, and the main calculation methods of Volterra kernel are the recursive method and identification method. The recursive algorithm is to substitute the relevant parameters of the system into the recursive equation to calculate the Volterra kernel. Billings and Tsang [12] adopted recursive algorithm to obtain Volterra kernels of each order for a nonlinear system. Jing et al. [13] established a description function from low-order Volterra kernel to high-order Volterra kernel to calculate the Volterra kernel of each order based on the nonlinear differential equation. Billings and Peyton Jones [14] adopted the harmonic probing method to derive the recursive algorithm of Volterra frequency-domain kernel. Swain and Billings [15] adopted recursive algorithm to calculate the Volterra kernel of an MIMO system. The research studies mentioned above calculated the Volterra kernel based on a system mathematical model, and the calculation accuracy depends entirely on the accuracy of the model. Obviously, recursive algorithm is not suitable for the system with uncertain mathematical model. The

identification algorithm is a “black box” operation, which does not rely on the mathematical model of the system; it only relies on the input and output data of the system. The Volterra kernel of each order can be calculated by adjusting the parameters according to the error between the true output and the estimated output. Toker and Emara-Shabaik [16] proposed LMS identification algorithm by using pseudorandom multilevel sequence excitation, which can avoid matrix inversion and improve calculation speed and accuracy. Sigrist et al. [17] adopted the improved LMS method to identify the first two-order Volterra time-domain kernel. Asyali and Juusola [18] expanded the Volterra kernel by the Meixner function to identify the Volterra model of a nonlinear system with time delay. Prazenica and Kurdila [19] proposed multiwavelet function to identify Volterra time-domain kernel, and the first three-order Volterra time-domain kernel function of a nonlinear system was obtained. The identification algorithm can realize the adaptive calculation, but this method calculated one Volterra kernel value at one time, which cannot achieve batch calculation and cannot meet the needs of fault diagnosis based on data. Therefore, there are at least two shortcomings in the traditional calculation of Volterra kernel: (1) the calculation efficiency and process are low and complex, respectively; (2) due to the approximate processing in the calculation process, the error between the estimated value and the true value is large.

In order to solve the problems of calculation methods mentioned above, a new intelligent calculation method is urgently needed. Artificial neural network (ANN) is a parallel distributed information processing network. The neurons in the network have strong ability of nonlinear approximation and fitting, which can approach any nonlinear function [20–22]. In recent years, to simplify Volterra kernel calculation, a small number of researchers have tried to adopt ANN to calculate the Volterra kernel. Wray and Green [23] and Stegmayer et al. [24] adopted the weight coefficient of ANN to express the Volterra kernel of each order, and it was also verified by a nonlinear system. In references [25, 26], a Volterra kernel identification method based on time-delay neural network (TDANN) was proposed, and the relationship between the parameters of network and Volterra kernel function was derived, which successfully identified the first three-order Volterra kernel of nonlinear unsteady aerodynamic loading. Although ANN adopted in above research studies can realize fast and accurate calculation of Volterra kernel, there are still some disadvantages, for example, the number of input nodes in the network structure was set by experience, and the analysis of input node number impact on the calculation accuracy was lacking; furthermore, the network structure was established with great randomness and arbitrariness, which has no adaptability.

The main contributions of this paper are as follows. (1) Based on the research studies by the authors of [23–26], the impact of input node number and the memory length of TDANN network on the accuracy and calculation speed is analyzed and studied. (2) Root mean square error (RMSE) is

introduced as the reference of neuron node number selection to realize network structure adaptive adjustment, which can avoid the network structure selection blindness. After obtaining the optimal structure, only the input and output of the system were needed to realize the intelligent calculation of the Volterra kernel. (3) The proposed method was verified by a nonlinear system. The reason why it was called “intelligent calculation” is that compared with the traditional calculation method, the whole calculation process does not need human’s participation, which can avoid the complex calculation process. In Section 2, the mathematical model of Volterra time-domain kernel is provided. Section 3 introduces the concept of TDANN. In Section 4, the relationship between the Volterra kernel model and TDANN model is analyzed, and the explicit expression of Volterra time-domain kernel is obtained. In Section 5, the proposed method is simulated and verified by a nonlinear model. The conclusions are summarized in Section 6.

2. Volterra Time-Domain Kernel Model

For any continuous time invariant nonlinear system, if the system input satisfies

$$\sqrt{\int_{-\infty}^{+\infty} u^2(t) dt} < +\infty, \quad (1)$$

then, the output of system can be expressed by Volterra series [27–29], namely,

$$y(t) = \sum_{n=1}^{+\infty} y_n(t), \quad (2)$$

$$y_n(t) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n u(t - \tau_i) d\tau_i, \quad (3)$$

where $y(t)$ is output of the system, $u(t)$ is input of the system, and $h_n(\tau_1, \tau_2, \dots, \tau_n)$ is the n -order Volterra time-domain kernel or generalized frequency response function. The n -order Volterra kernel model can be obtained by truncating (2), which is shown as

$$\begin{aligned} y(t) = & h_0 + \int_0^{+\infty} h_1(\tau_1) u(t - \tau_1) d\tau_1 \\ & + \int_0^{+\infty} \int_0^{+\infty} h_2(\tau_1, \tau_2) u(t - \tau_1) u(t - \tau_2) d\tau_1 d\tau_2 \\ & + \cdots + \int_0^{+\infty} \int_0^{+\infty} \int_0^{+\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) u(t - \tau_1) \\ & \cdot u(t - \tau_2) \cdots u(t - \tau_n) d\tau_1 d\tau_2 \cdots d\tau_n. \end{aligned} \quad (4)$$

Discretize equation (3) to obtain the discrete Volterra kernel model, which is shown as

$$\begin{aligned}
y(k) &= h_0 + \sum_{m_1=0}^{+\infty} h_1(m_1)u(k-m_1) \\
&+ \sum_{m_1=0}^{+\infty} \sum_{m_2=0}^{+\infty} h_2(m_1, m_2)u(k-m_1)u(k-m_2) + \dots \\
&+ \sum_{m_1=0}^{+\infty} \sum_{m_2=0}^{+\infty} \dots \sum_{m_n=0}^{+\infty} h_n(m_1, m_2, \dots, m_n) \\
&\cdot u(k-m_1)u(k-m_2).
\end{aligned} \tag{5}$$

For a system with memory length M , equation (4) can be changed to

$$\begin{aligned}
y(k) &= h_0 + \sum_{m_1=0}^M h_1(m_1)u(k-m_1) \\
&+ \sum_{m_1=0}^M \sum_{m_2=0}^M h_2(m_1, m_2)u(k-m_1)u(k-m_2) + \dots \\
&+ \sum_{m_1=0}^M \sum_{m_2=0}^M \dots \sum_{m_n=0}^M h_n(m_1, m_2, \dots, m_n) \\
&\cdot u(k-m_1)u(k-m_2) \dots u(k-m_n).
\end{aligned} \tag{6}$$

Generally speaking, the energy components of Volterra kernels above three-order are very small, and it is also not easy to visualize the graphs over three dimensions [30]. Therefore, this paper only calculates the first three-order Volterra kernel of the system.

3. Time-Delay Artificial Neural Networks (TDANNs)

As shown in Figure 1, the artificial neural network is composed of many neurons which are related to each other to achieve efficient data processing. Such a network is also known as feedforward multilayer perceptron. Artificial neural network is generally composed of an input layer, hidden layer, and output layer. A simple three-layer network can realize the approximation of any nonlinear function. A mapping of data from the input layer to the output layer is called training. In order to achieve accurate approximation, several training iterations are required. In this process, the network constantly updates the weights and thresholds between different neurons. According to the error between true value and calculation result, the weights and thresholds are modified in turn. After updating several times, the calculation result is closer to the true value, and the optimal weight and threshold can also be obtained.

ANN is a typical MIMO neural network with M inputs and N outputs. The selection of input and output of a neural network depends on the characteristics of the specific system. For the Volterra kernel model, from equation (5), it can be seen that the Volterra kernel model with memory length M has $M+1$ input variables, which are $u(k), u(k-1), \dots, u(k-M)$, respectively. There is one

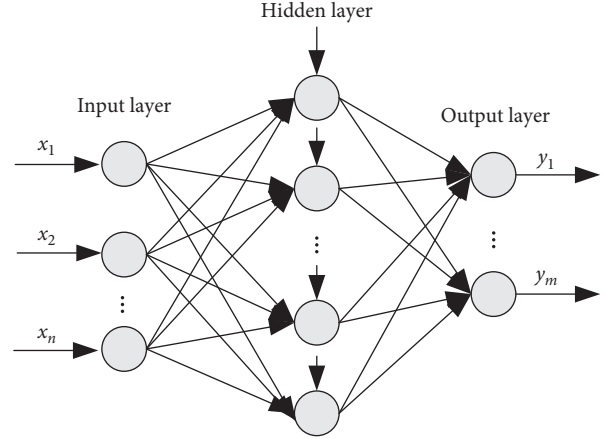


FIGURE 1: Artificial neural network with single hidden layer.

output, that is $y(k)$. In addition, $h(k-1), \dots, u(k-M)$ are delayed $1t, 2t, \dots, Mt$, respectively, from input $u(k)$. For a single-input single-output (SISO) system, how to construct dataset with $M+1$ inputs and 1 output is important. The specific process is shown in Figure 2.

The artificial neural network takes $M+1$ delay neurons as the input layer and 1 output neuron as the output layer, which is called time-delay artificial neural network (TDANN). Because the network adopts forward time delay, it has the function of memory. The structure of TDANN is shown in Figure 3.

In Figure 3, $\omega_{11}, \varphi_{12}, \dots, \omega_{Mp}$ are weights between the input layer and hidden layer; b_1, b_2, \dots, b_p are thresholds on the hidden layer; $\omega_1, \omega_2, \dots, \omega_p$ are weights between the hidden layer and output layer; and b_0 is the threshold on the output layer.

4. Equation Derivation of Volterra Kernel Based on TDANN

According to equation (3), the first three-order Volterra kernel model with memory length M can be shown as

$$\begin{aligned}
y(k) &= h_0 + \sum_{m_1=0}^M h_1(m_1)u(k-m_1) \\
&+ \sum_{m_1=0}^M \sum_{m_2=0}^M h_2(m_1, m_2)u(k-m_1)u(k-m_2) \\
&+ \sum_{m_1=0}^M \sum_{m_2=0}^M \sum_{m_3=0}^M h_n(m_1, m_2, m_3)u(k-m_1) \\
&\cdot u(k-m_2)u(k-m_3).
\end{aligned} \tag{7}$$

For the TDANN model shown in Figure 3, the selection of activation function and number of neurons in the hidden layer directly affects the accuracy of output. In this paper, *sigmoid* function is chosen as the activation function of neurons in the hidden layer, which is shown as

$$\varphi(x) = \frac{1}{1 + e^{-x}}. \tag{8}$$

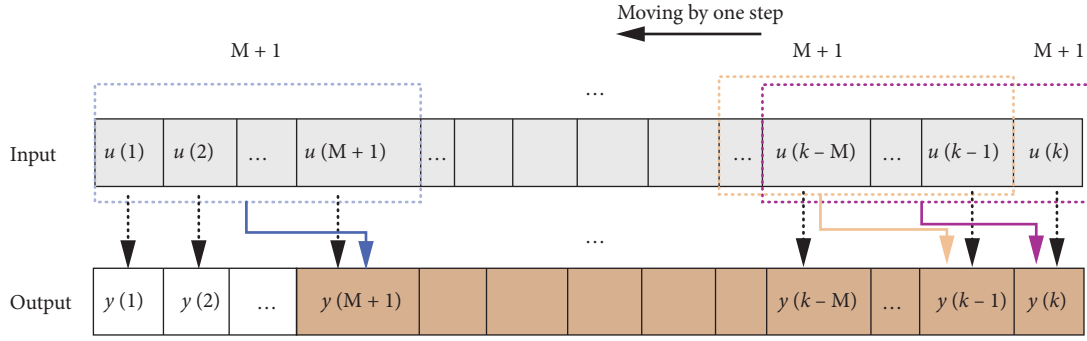


FIGURE 2: The process of dataset construction.

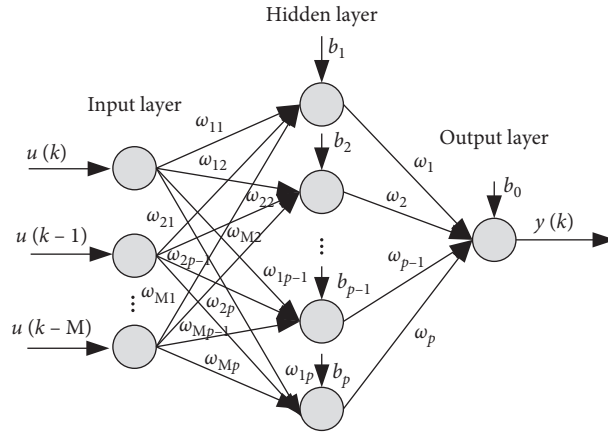


FIGURE 3: The structure of TDANN.

The number of hidden neurons has a significant impact on the performance of the network. If the number of neurons is too small, the calculation accuracy is low. If the number of neurons is too large, the overfitting phenomena will occur. So, according to the Volterra kernel characteristics, this paper adopts root mean square error (RMSE) as an evaluation index to determine the reasonable number of hidden neurons, which is shown as

$$\text{RMSE} = \sqrt{\frac{1}{k} \sum_{j=1}^k [y(j) - \hat{y}(j)]^2}, \quad (9)$$

where $y(j)$ is the true value of output at time j ; $\hat{y}(j)$ is the predicted value of output at time j ; and RMSE represents the standard deviation of error samples between the estimated value and the true value, which reflects the dispersion degree of sample. The smaller the RMSE is, the better the approximation effect is.

$\varphi(x)$ is chosen as activation function between neurons in the input layer and hidden layer, and *pureline* function is chosen as activation function between neurons in the hidden layer and output layer; then, the j -th output y_j can be expressed as

$$y_j = \varphi \left(\sum_{i=1}^{M+1} \omega_{ij} * u_i + b_j \right), \quad (10)$$

where M is the memory length of the network; ω_{ij} is the weight between the i -th neuron in the input layer and the j -th neuron in the hidden layer; u_i is the i -th input; and b_j is the threshold of j -th neuron in the hidden layer.

The output of TDANN can be expressed by

$$y_{\text{out}} = \sum_{j=1}^p \omega_j * y_j + b_0, \quad (11)$$

where p is the number of neurons, ω_j is the weight between the j -th neuron in the hidden layer and output layer, and b_0 is the threshold on the output layer. Substituting equation (9) into equation (10) and expanding $\varphi(x)$ with Taylor series, equation (12) can be obtained.

$$y_{\text{out}} = \sum_{j=1}^p \omega_j * \sum_{n=0}^{+\infty} \frac{\varphi^{(n)}(b_j)}{n!} * \left[\sum_{i=1}^{M+1} \omega_{ij} * u(k - (i - 1)) \right]^n + b_0, \quad (12)$$

where $\varphi^{(n)}(b_j)$ stands for the n -th derivative of $\varphi(x)$ at b_j . Because only the first three-order Volterra kernel values are considered in this paper, equation (11) can be translated into

$$y_{\text{out}} = \sum_{j=1}^p \omega_j * \sum_{n=0}^3 \frac{\varphi^{(n)}(b_j)}{n!} * \left[\sum_{i=1}^{M+1} \omega_{ij} * u(k - (i - 1)) \right]^n + b_0. \quad (13)$$

Expanding n and i in equation (13), equation (14) can be obtained.

$$\begin{aligned}
y_{\text{out}} = & b_0 + \sum_{j=1}^p \omega_j * \varphi(b_j) + \sum_{j=1}^p \omega_j * \varphi^{(1)}(b_j) [\omega_{1j} * u(k) + \omega_{2j} * u(k-1) + \dots + \omega_{M+1j} * u(k-M)] \\
& + \sum_{j=1}^p \omega_j * \frac{\varphi^{(2)}(b_j)}{2!} [\omega_{1j} * u(k) + \omega_{2j} * u(k-1) + \dots + \omega_{M+1j} * u(k-M)]^2 \\
& + \sum_{j=1}^p \omega_j * \frac{\varphi^{(3)}(b_j)}{3!} [\omega_{1j} * u(k) + \omega_{2j} * u(k-1) + \dots + \omega_{M+1j} * u(k-M)]^3.
\end{aligned} \tag{14}$$

Comparing equation (6) and equation (14), the first three-order Volterra time-domain kernel calculation equations can be obtained, which are shown as follows:

$$h_0 = b_0 + \sum_{j=1}^p \omega_j * \varphi(b_j), \tag{15}$$

$$h_1(m) = \sum_{j=1}^p \omega_j * \omega_{m+1j} * \varphi^{(1)}(b_j) \quad m = 0, 1, 2, \dots, M, \tag{16}$$

$$h_2(m_1, m_2)_{m_1 \neq m_2} = \sum_{j=1}^p \omega_j * \omega_{m_1+1j} * \omega_{m_2+1j} * \varphi^{(2)}(b_j) \quad m_1 = 0, 1, 2, \dots, M; m_2 = 0, 1, 2, \dots, M, \tag{17}$$

$$h_2(m_1, m_2)_{m_1 = m_2} = \sum_{j=1}^p \omega_j * (\omega_{m_1+1j})^2 * \frac{\varphi^{(2)}(b_j)}{2!} \quad m_1 = 0, 1, 2, \dots, M; m_2 = 0, 1, 2, \dots, M, \tag{18}$$

$$h_3(m_1, m_2, m_3)_{m_1 \neq m_2 \neq m_3} = \sum_{j=1}^p \omega_j * \omega_{m_1+1j} * \omega_{m_2+1j} * \omega_{m_3+1j} * \varphi^{(3)}(b_j) \quad m_1 = 0, 1, 2, \dots, M; m_2 = 0, 1, 2, \dots, M; m_3 = 0, 1, 2, \dots, M, \tag{19}$$

$$h_3(m_1, m_2, m_3)_{m_1 = m_2 = m_3} = \sum_{j=1}^p \omega_j * (\omega_{m_1+1j})^3 * \frac{\varphi^{(3)}(b_j)}{3!} \quad m_1 = 0, 1, 2, \dots, M; m_2 = 0, 1, 2, \dots, M; m_3 = 0, 1, 2, \dots, M, \tag{20}$$

$$h_3(m_1, m_2, m_3)_{\substack{m_1 = m_3 \neq m_2 \\ m_1 = m_2 \neq m_3 \\ m_1 \neq m_2 = m_3}} = \sum_{j=1}^p \omega_j * \omega_{m_1+1j} * \omega_{m_2+1j} * \omega_{m_3+1j} * \frac{\varphi^{(3)}(b_j)}{2} \quad m_1 = 0, 1, 2, \dots, M; m_2 = 0, 1, 2, \dots, M; m_3 = 0, 1, 2, \dots, M. \tag{21}$$

It can be seen from equations (15)–(21) that the first three-order Volterra time-domain kernel expressions are completely composed of weights and thresholds of TDANN. We substitute the optimal weights and thresholds into the equations above to calculate the Volterra kernel of each

order. Compared with the traditional calculation method, the new method only needs the input and output of system to train the network and does not need manual participation, which can realize the intelligent calculation of Volterra kernel.

5. Simulations

According to reference [31], the dynamic model of the nonlinear oscillator is shown in Figure 4.

The nonlinear differential equation of the model can be expressed as follows:

$$m \frac{d^2 y(t)}{dt^2} + c \frac{dy(t)}{dt} + k_1 y(t) + k_2 y^2(t) + k_3 y^3(t) = u(t), \quad (22)$$

where m is the mass of the oscillator; c is the damping coefficient of the system; k_1 and k_2 are linear stiffness parameters and k_3 is nonlinear stiffness parameter; $u(t)$ is external force applied to the system, which is the input of the system; and $y(t)$ is the displacement of the vibrator, which is the output of the system.

The parameters are

$$m_1 = 1\text{kg}, c = \frac{3\text{N} \cdot \text{s}}{\text{m}}, k_1 = \frac{60\text{N}}{\text{m}}, k_2 = \frac{100\text{N}}{\text{m}^2}, k_3 = \frac{200\text{N}}{\text{m}^3}, \quad (23)$$

setting the input as $u(t) = 50\cos(0.4\pi t)$.

5.1. The Effect of Volterra Memory Length M on System Output. From Figures 2 and 3, it can be seen that the memory length M not only affects the neuron number of the input layer but also affects the composition of dataset. If M is too small, the nonlinear correlation between neurons is not high. If M is too large, overfitting phenomena may occur. So, it is very important to choose M reasonably to ensure the output accuracy. In order to facilitate the analysis, this paper makes the neuron number of the input layer equal to the hidden layer, namely, $M + 1 = p$. The influences of M on the network output are tested, which are shown in Figures 5–7 and Table 1.

The specific effects of memory length M on system output are shown in Table 1.

From Figures 5–7 and Table 1, it can be seen that M has a great effect on the accuracy of system output. At the beginning, when M is gradually increasing, the system output error tends to decrease due to the increase of correlation

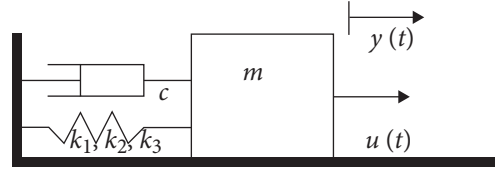


FIGURE 4: The model of the nonlinear oscillator.

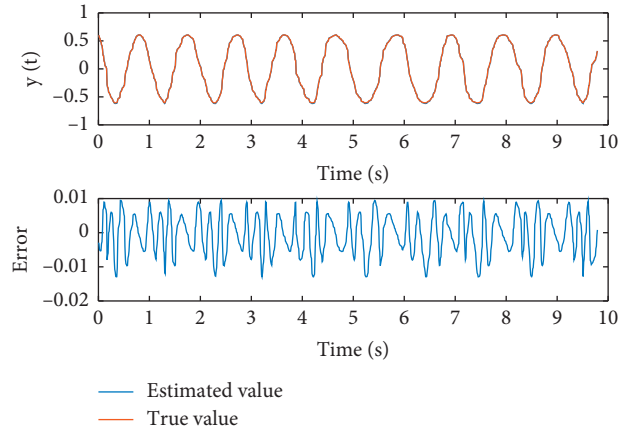


FIGURE 5: $M = 3$: the comparison of output between true value and estimated value.

degree between neurons in the network. However, with the increase of M , the degree of correlation between neurons is increasing, but the overfitting phenomena occurred, so the system output error shows oscillation. In addition, with the increase of M , the structure of network becomes more complex, and the time cost of network training and testing is longer. Furthermore, based on the RMSE index, the memory length M is selected to be 8.

5.2. The Calculation of Volterra Kernel. When $M=8$, the neuron number of the input layer is 9, and the neuron number of the hidden layer is 9. After TDANN training, the optimal weight of different neurons between the input layer and hidden layer is

$$\omega_{i,j} = \begin{bmatrix} 0.4991 & 0.4327 & 0.4583 & 0.4240 & 0.4035 & 0.3351 & 0.2906 & 0.3192 & 0.2828 \\ 0.0116 & 0.0118 & 0.0199 & 0.0443 & 0.0349 & 0.0092 & -0.0498 & -0.0302 & -0.0562 \\ -0.2804 & -0.3477 & -0.3369 & -0.3534 & -0.4014 & -0.3974 & -0.4240 & -0.4185 & -0.4601 \\ -0.1466 & -0.1077 & -0.0415 & 0.0362 & 0.0556 & 0.1175 & 0.1556 & 0.1616 & 0.2020 \\ 3.6585 & 2.6603 & 1.6866 & 0.7550 & -0.2669 & -1.2297 & -2.2024 & -3.1543 & -4.0925 \\ -0.7376 & -0.5524 & -0.3880 & -0.1766 & 0.0106 & 0.1992 & 0.3593 & 0.5723 & 0.7083 \\ 0.2753 & 0.2306 & 0.1639 & 0.0704 & 0.0402 & -0.0262 & -0.1491 & -0.1554 & -0.2414 \\ 0.0340 & 0.0828 & 0.0036 & 0.0095 & 0.0146 & -0.0228 & -0.0451 & -0.0031 & -0.0656 \\ 0.0649 & 0.0233 & 0.0389 & -0.0084 & -0.0120 & -0.0324 & -0.0195 & 0.0140 & 0.0064 \end{bmatrix}. \quad (24)$$

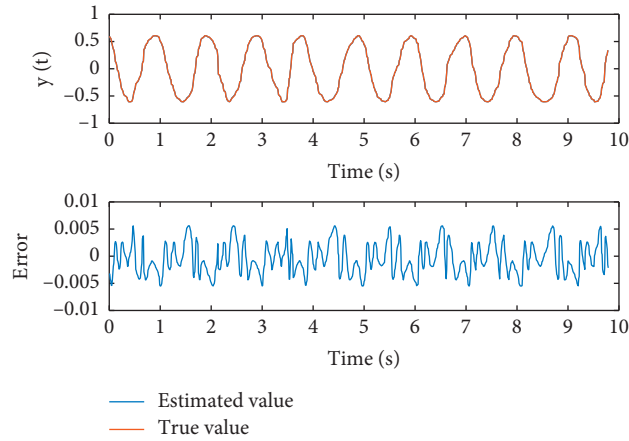


FIGURE 6: $M = 8$: the comparison of output between true value and estimated value.

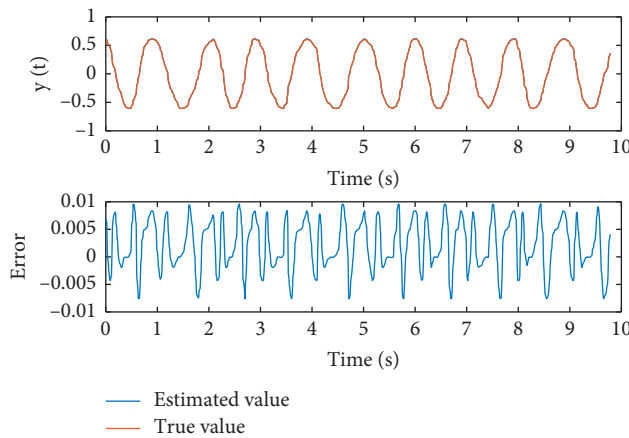


FIGURE 7: $M = 10$: the comparison of output between true value and estimated value.

The optimal threshold of neurons in the hidden layer is

$$b = [-2.4254 \quad 2.6970 \quad -2.4143 \quad 1.0530 \quad 0.0096 \quad 0.7809 \quad 1.9744 \quad -2.7337 \quad 3.5764]. \quad (25)$$

The optimal weight of different neurons between the hidden layer and output layer is

$$\omega_j = [0.3457 \quad -0.0424 \quad -0.3568 \quad -0.1451 \quad -1.8627 \quad -1.1851 \quad 0.1561 \quad 0.6817 \quad 0.4550]. \quad (26)$$

The optimal threshold of neurons in the output layer is

$$b_0 = 1.2744. \quad (27)$$

The first three-order Volterra time-domain kernel can be obtained by substituting the parameters above into equations (15)–(20), and the results are shown in Figures 8–10.

In order to prove that the identified kernels hold the correct dynamics, predicted responses can be obtained by

the first three-order Volterra time-domain kernels, which are shown in Figure 11.

From Figure 11, it can be seen that the predicted response using first-order kernel is similar to the system input, which stands for the linear system response, and the predicted response using first two-order kernels begin to appear as nonlinearity components; furthermore, with the order of kernel increasing, the nonlinear components are more and more obvious and also close to the real response.

TABLE 1: The effects of M on system output.

| M | Average error | RMSE | Time cost (s) |
|-----|---|---------------|---------------|
| 2 | 0.0013 | 0.006 | 0.91 |
| 3 | 6.3360×10^{-4} | 0.0059 | 0.85 |
| 4 | 8.5099×10^{-4} | 0.0051 | 0.90 |
| 5 | 6.1712×10^{-4} | 0.0042 | 0.79 |
| 6 | 0.0033 | 0.0052 | 0.96 |
| 7 | 1.3466×10^{-4} | 0.0052 | 0.89 |
| 8 | 8.7882×10^{-5} | 0.0013 | 1.02 |
| 9 | 8.3969×10^{-4} | 0.0031 | 1.24 |
| 10 | 0.0021 | 0.0050 | 1.29 |

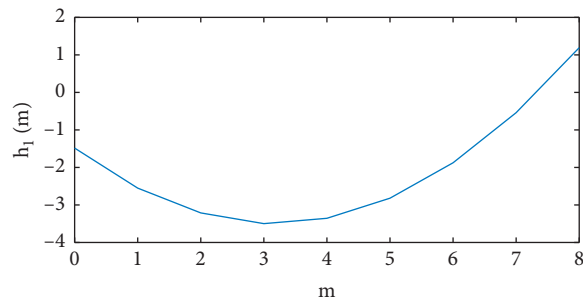


FIGURE 8: The first-order Volterra time-domain kernel calculated by TDANN.

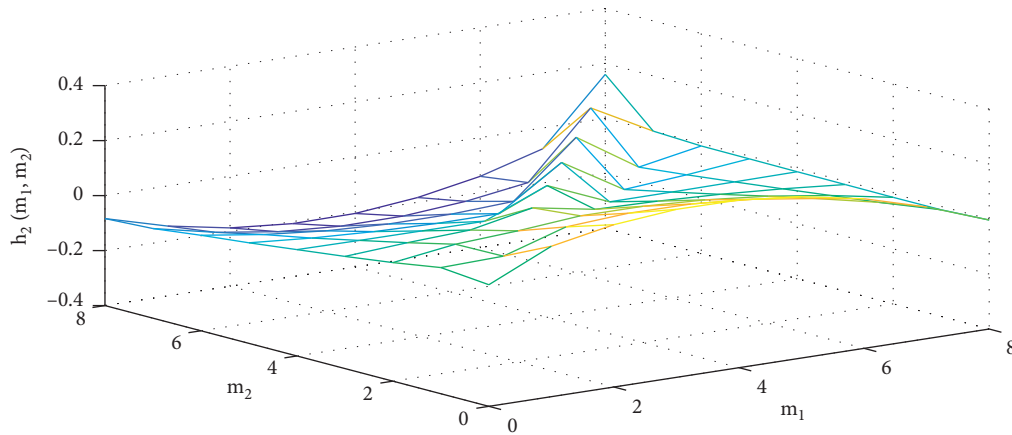


FIGURE 9: The second-order Volterra time-domain kernel calculated by TDANN.

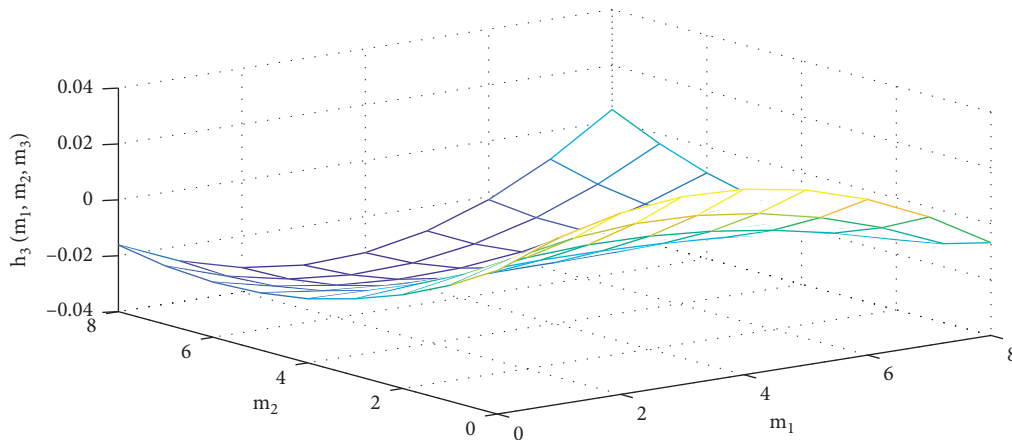


FIGURE 10: The third-order Volterra time-domain kernel calculated by TDANN ($m_3 = 0$).

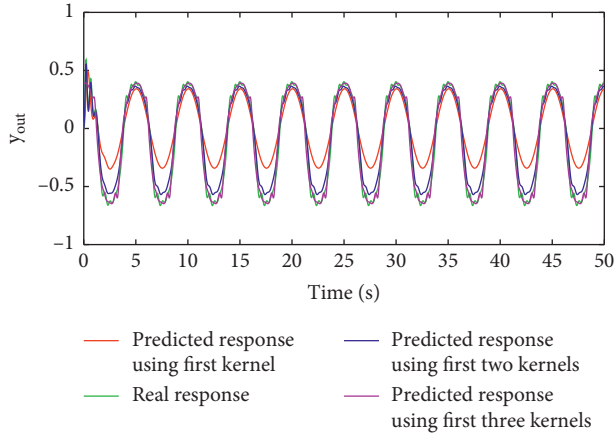


FIGURE 11: The predicted response of each assessed kernel.

5.3. *The Robustness of the Proposed Method.* In order to demonstrate the robustness of the proposed method, Gaussian noise is added to the system input; the first three-order Volterra kernels are calculated by TDANN, and then the system output estimated value can be obtained by equation (6), which is shown in Figure 12.

From Figure 12, it can be seen that due to the influence of noise, the estimated output value of the system is in fluctuation state at beginning, and the errors between the estimated value and the true value are relatively large, but after about 2 s, the estimated value begins to approach the true value, and after 4.7 s, the error between the estimated value and the true value is getting smaller and smaller. So, it can be demonstrated that the proposed method has robustness.

5.4. *The Accuracy Comparison with Traditional Algorithm.* In order to verify the accuracy of Volterra kernel calculated by TDANN, the first three-order Volterra time-domain kernels of the nonlinear system are calculated by TDANN, recursive algorithm in reference [32], and variable step size least mean square (VSSLMS) adaptive identification algorithm in reference [33], respectively. Then, three different output results are obtained by equation (6). The accuracy of three methods is shown in Figures 13 and 14.

From Figures 13 and 14, it can be seen that among three methods, the accuracy of the TDANN method is highest, so each order of Volterra time-domain kernel (as shown in Figures 7–9) calculated by TDANN is closest to the true value of Volterra time-domain kernel. In the VSSLMS identification algorithm, all kinds of parameters have been adjusted to the best, but the identification accuracy is not ideal for the system; two reasons can account for such phenomena. Firstly, the adjustment of parameters in the identification algorithm is based on experience, which is subjective. Secondly, the accuracy of the identification algorithm may be affected by the data type. So, the accuracy of identification algorithm is the lowest. It is worth to notice that in order to further enhance the accuracy, TDANN can easily realize the above three-order Volterra time-domain

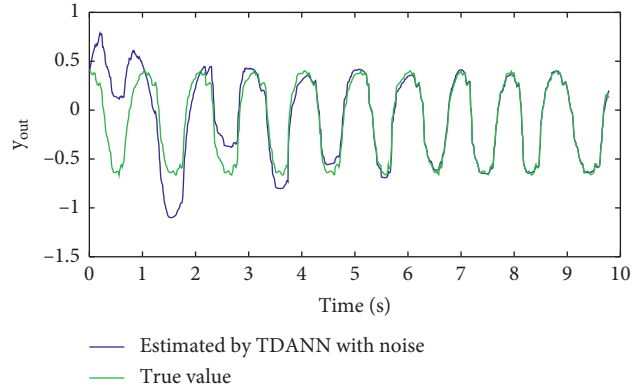


FIGURE 12: The robustness of the proposed method.

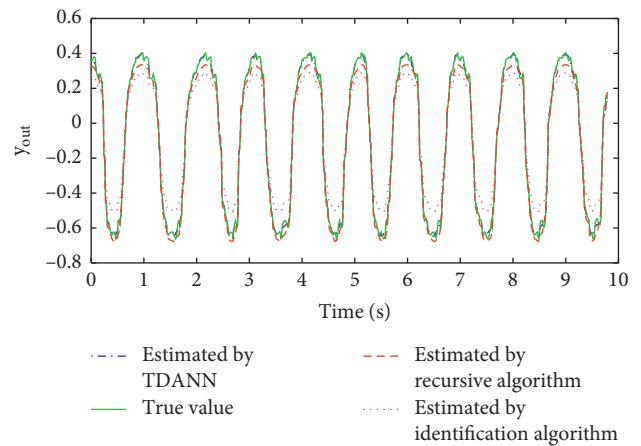


FIGURE 13: The output comparison of three methods.

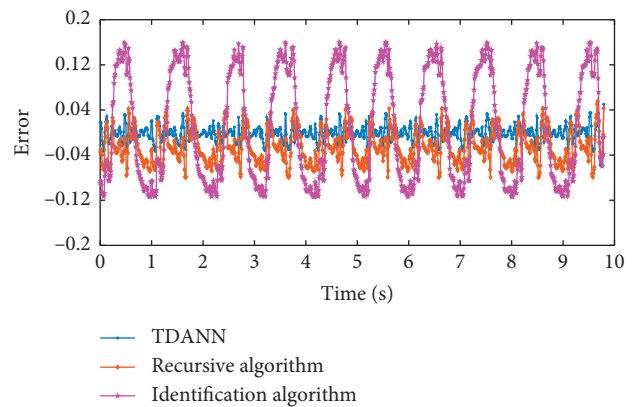


FIGURE 14: The errors of three methods.

kernel calculation, while it is difficult for others to calculate the Volterra time-domain kernel above three-order.

5.5. *The Computation Complexity Comparison with Traditional Algorithm.* Both TDANN and recursive algorithm can realize the simultaneous calculation of multiple Volterra kernels, namely, one operation can realize the

TABLE 2: The time consumption of different methods.

| Samples | Identification algorithm (ms) | Recursive algorithm (ms) | TDANN (ms) |
|---------|-------------------------------|--------------------------|------------|
| 2 | 113.34 | 185.69 | |
| 4 | 183.47 | | 145.31 |
| 6 | 124.28 | | |
| (2,3) | 129.35 | 194.16 | |
| (4,7) | 175.48 | | 117.48 |
| (5,5) | 116.72 | | |
| (1,4,0) | 163.89 | 179.35 | |
| (4,7,0) | 142.51 | | 104.17 |
| (5,6,0) | 176.19 | | |

calculation of multiple Volterra kernels, which can improve the calculation speed and efficiency, while with identification algorithm, one operation can only calculate one Volterra kernel value. In order to demonstrate that TDANN can reduce the computational complexity, the recursive algorithm in reference [32] and the identification algorithm in reference [33] are adopted to randomly calculate several first three-order Volterra kernels of the system. The time consumption of three methods is shown in Table 2.

It can be seen that the identification algorithm consumed the most time, and TDANN consumed the least time. The time consumption of identification algorithm, recursive algorithm, and TDANN is 1325.25 ms, 559.2 ms, and 366.96 ms, respectively. Compared with identification algorithm and recursive algorithm, the calculation speed of TDANN is increased by 72.31% and 34.38%, respectively. Therefore, the method proposed in this paper reduced the complexity of calculation, which can realize the online calculation and detection of the system.

6. Conclusions

To solve the problems of large computation amount and low accuracy in the conventional method to calculate the Volterra time-domain kernel of the nonlinear system, this paper proposed a novel method to intelligently calculate Volterra time-delay kernel by TDANN; by comparing the Volterra kernel model with the TDANN model, the connection of the two models has been found, and the complex Volterra time-domain kernel calculation was transformed into the optimal parameter calculation of neural network; the novel method has been well verified on a nonlinear system.

The main achievements can be summarized as follows.

Compared with recursive algorithm, the TDANN method has three contributions: Firstly, it can avoid the complex modeling operation of the system; only with input and output of the system, the first three-order Volterra time-domain kernel can be calculated, so it is more adaptable. Secondly, it transforms the complex calculation into the optimal parameter solution of the network, and the Volterra time-domain kernel can be calculated through network training, so it is more intelligent. Thirdly, it can adjust the network structure according to the output of the network to

ensure the calculation result is closer to the true value, so it is more adaptive.

Compared with the identification method, the TDANN method can not only achieve batch calculation at one time but also reduce the time consumption, so the calculation efficiency is high, which provides the possibility of online calculation.

However, the three-order Volterra kernels calculated by TDANN are only parts of the whole Volterra kernel information because the value of M cannot be too large. How to balance the amount of Volterra kernel information and time consumption is the future work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Shaanxi Key Industry Innovation Chain Project (no. 2019ZDLGY01-01-02) and Foshan Key Science and Technology Project (no. 2016AG101813).

References

- [1] C. M. Cheng, Z. K. Peng, X. J. Dong, W. M. Zhang, and G. Meng, "Nonlinear system identification using Kautz basis expansion-based Volterra-PARAFAC model," *Nonlinear Dynamics*, vol. 94, no. 3, pp. 2277–2287, 2018.
- [2] J. C. P. Jones and K. S. A. Yaser, "A new harmonic probing algorithm for computing the MIMO Volterra frequency response functions of nonlinear systems," *Nonlinear Dynamics*, vol. 94, no. 2, pp. 1029–1046, 2018.
- [3] H. T. Han, H. G. Ma, L. N. Tan, J. F. Cao, and J. L. Zhang, "Non-parametric identification method of Volterra kernels for nonlinear systems excited by multitone signal," *Asian Journal of Control*, vol. 16, no. 2, pp. 519–529, 2014.
- [4] Y. Zhu and Z. Q. Lang, "Design of nonlinear systems in the frequency domain: an output frequency response function-based approach," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1358–1371, 2018.
- [5] P. Li, Z. Q. Lang, L. Zhao et al., "System identification-based frequency domain feature extraction for defect detection and characterization," *NDT & E International*, vol. 98, pp. 70–79, 2018.
- [6] T. Wu and A. Kareem, "Vortex-induced vibration of bridge decks: Volterra series-based model," *Journal of Engineering Mechanics*, vol. 139, no. 12, pp. 1831–1843, 2013.
- [7] J. Zhang, J. Cao, and F. Gao, "Fault diagnosis for multivariable non-linear systems based on non-linear spectrum feature," *Transactions of the Institute of Measurement and Control*, vol. 39, no. 7, pp. 1017–1026, 2017.
- [8] X. Xia, J. Zhou, C. Li, and W. Zhu, "A novel method for fault diagnosis of hydro generator based on NOFRFs," *International Journal of Electrical Power & Energy Systems*, vol. 71, pp. 60–67, 2015.

- [9] Y. Zhu and Z. Q. Lang, "The effects of linear and nonlinear characteristic parameters on the output frequency responses of nonlinear systems: the associated output frequency response function," *Automatica*, vol. 93, pp. 422–427, 2018.
- [10] H. Ma, J. Zeng, Z. Lang, L. Zhang, Y. Guo, and B. Wen, "Analysis of the dynamic characteristics of a slant-cracked cantilever beam," *Mechanical Systems and Signal Processing*, vol. 75, pp. 261–279, 2016.
- [11] W. Yang, Z. Lang, and W. Tian, "Condition monitoring and damage location of wind turbine blades by frequency response transmissibility analysis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6558–6564, 2015.
- [12] S. A. Billings and K. M. Tsang, "Spectral analysis for nonlinear systems, Part I: parametric non-linear spectral analysis," *Mechanical Systems and Signal Processing*, vol. 3, no. 4, pp. 319–339, 1989.
- [13] X. J. Jing, Z. Q. Lang, and S. A. Billings, "Mapping from parametric characteristics to generalized frequency response functions of non-linear systems," *International Journal of Control*, vol. 81, no. 7, pp. 1071–1088, 2008.
- [14] S. A. Billings and J. C. Peyton Jones, "Mapping non-linear integro-differential equations into the frequency domain," *International Journal of Control*, vol. 52, no. 4, pp. 863–879, 1990.
- [15] A. K. Swain and S. A. Billings, "Generalized frequency response function matrix for MIMO non-linear systems," *International Journal of Control*, vol. 74, no. 8, pp. 829–844, 2001.
- [16] O. Toker and H. E. Emara-Shabaik, "Pseudo-random multilevel sequences: spectral properties and identification of Hammerstein systems," *IMA Journal of Mathematical Control and Information*, vol. 21, no. 2, pp. 183–205, 2004.
- [17] Z. Sigrist, E. Grivel, and B. Alcoverro, "Estimating second-order Volterra system parameters from noisy measurements based on an LMS variant or an errors-in-variables method," *Signal Processing*, vol. 92, no. 4, pp. 1010–1020, 2012.
- [18] M. H. Asyali and M. Juusola, "Use of Meixner functions in estimation of Volterra kernels of nonlinear systems with delay," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 2, pp. 229–237, 2005.
- [19] R. J. Prazenica and A. J. Kurdila, "Multiwavelet constructions and Volterra kernel identification," *Nonlinear Dynamics*, vol. 43, no. 3, pp. 277–310, 2006.
- [20] B. Sahoo, S. K. Routray, and P. K. Rout, "Artificial neural network-based PI-controlled reduced switch cascaded multilevel inverter operation in wind energy conversion system with solid-state transformer," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 43, no. 4, pp. 1053–1073, 2019.
- [21] F. Shaikat, G. Raja, R. Ashraf, S. Khalid, M. Ahmad, and A. Ali, "Artificial neural network based classification of lung nodules in CT images using intensity, shape and texture features," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 10, pp. 4135–4149, 2019.
- [22] X. Han, H. Xiang, Y. Li, and Y. Wang, "Predictions of vertical train-bridge response using artificial neural network-based surrogate model," *Advances in Structural Engineering*, vol. 22, no. 12, pp. 2712–2723, 2019.
- [23] J. Wray and G. G. R. Green, "Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network," *Biological Cybernetics*, vol. 71, no. 3, pp. 187–195, 1994.
- [24] G. Stegmayer, M. Pirola, G. Orengo, and O. Chiotti, "Towards a Volterra series representation from a Neural Network model," *WSEAS Transactions on Systems*, vol. 3, no. 2, pp. 432–437, 2004.
- [25] N. C. G. De Paula and F. D. Marques, "Multi-variable Volterra kernels identification using time-delay neural networks: application to unsteady aerodynamic loading," *Nonlinear Dynamics*, vol. 97, no. 1, pp. 767–780, 2004.
- [26] N. C. G. De Paula, F. D. Marques, and W. A. Silva, "Volterra kernels assessment via time-delay neural networks for nonlinear unsteady aerodynamic loading identification," *AIAA Journal*, vol. 57, no. 4, pp. 1–11, 2018.
- [27] B. Zhang and S. A. Billings, "Volterra series truncation and kernel estimation of nonlinear systems in the frequency domain," *Mechanical Systems and Signal Processing*, PR, vol. 84, pp. 39–57, 2017.
- [28] R. S. Bayma, Y. Zhu, and Z.-Q. Lang, "The analysis of nonlinear systems in the frequency domain using nonlinear output frequency response functions," *Automatica*, vol. 94, pp. 452–457, 2018.
- [29] X. Jing, Z. Lang, and S. A. Billings, "Determination of the analytical parametric relationship for output spectrum of Volterra systems based on its parametric characteristics," *Journal of Mathematical Analysis and Applications*, vol. 351, no. 2, pp. 694–706, 2009.
- [30] J. Misić, V. Marković, and Z. Marinković, *Volterra Kernels Extraction From Neural Networks For Amplifier Behavioral Modeling*, vol. 27–29, BIHTEL, Sarajevo, Bosnia and Herzegovina.
- [31] Z. Q. Lang * and S. A. Billings, "Energy transfer properties of non-linear systems in the frequency domain," *International Journal of Control*, vol. 78, no. 5, pp. 345–362, 2005.
- [32] J. C. P. Jones and S. A. Billings, "Recursive algorithm for computing the frequency response of a class of non-linear difference equation models," *International Journal of Control*, vol. 50, no. 5, pp. 1925–1940, 1989.
- [33] L. R. Chen, Z. R. Zhang, and J. F. Cao, "A novel method of combining generalized frequency response function and convolutional neural network for complex system fault diagnosis," *Plos One*, vol. 15, no. 2, Article ID e0228324, 2020.