

Article

Self-Evolving Chebyshev Radial Basis Function Neural Complementary Sliding Mode Control

Lei Zhang ¹, Xiangguo Li ² and Juntao Fei ^{1,2,3,*}¹ College of Information Science and Engineering, Hohai University, Changzhou 213022, China² College of Mechanical and Electrical Engineering, Hohai University, Changzhou 213022, China³ Jiangsu Key Laboratory of Power Transmission and Distribution Equipment Technology, College of Artificial Intelligence and Automation, Hohai University, Changzhou 213022, China

* Correspondence: jtfei@hhu.edu.cn

Abstract: A novel intelligent complementary sliding mode control (ICSMC) method is proposed for nonlinear systems with unknown uncertainties in this paper. A self-evolving Chebyshev radial basis function neural network (RBFNN) (SECRBFNN) with self-learning parameters and structure is proposed and combined with complementary sliding mode control (CSMC). CSMC not only has the advantages of the strong robustness of traditional SMC but also has certain advantages in reducing chattering and control accuracy. The SECRBFNN, which combines the advantages of the Chebyshev network (CN) and an RBFNN, is used to estimate unknown uncertainties in nonlinear systems. Meanwhile, a node self-evolution mechanism is proposed to avoid redundancy in the number of neurons. Eventually, the detailed simulation results demonstrate the feasibility and superiority of the proposed method.

Keywords: complementary sliding mode control (CSMC); self-evolving Chebyshev radial basis function neural network (SECRBFNN); node self-evolution mechanism; nonlinear systems control

MSC: 68T07; 93C40



Citation: Zhang, L.; Li, X.; Fei, J. Self-Evolving Chebyshev Radial Basis Function Neural Complementary Sliding Mode Control. *Mathematics* **2023**, *11*, 3231. <https://doi.org/10.3390/math11143231>

Academic Editor: Fernando Gaxiola

Received: 9 June 2023

Revised: 12 July 2023

Accepted: 20 July 2023

Published: 22 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the practical application of control technology, there is a general difference between the actual model and the known nominal model. This difference makes it difficult to design controllers for a group of nonlinear systems [1,2]. In past research, some linear control methods have been applied to solve the control problems of unknown models, but poor control accuracy and overall performance are the disadvantages of these methods. In addition, some model-independent control methods such as neural network control [3], fuzzy logic systems [4,5], and PID control are also applied to unknown systems [6]. However, the parameter adjustment of these methods needs to be combined with actual application scenarios, which means that empirical methods are almost the only methods to adjust parameters in the face of different unknown models.

Sliding mode control (SMC), as a control method which does not depend on an accurate model and has strong robustness, has unique advantages in solving the control field of systems with unknown uncertainties [7]. In many applications, SMC is often used to suppress the influence of lumped uncertainty in the system and achieve fast response of control [8]. However, the discontinuous switching characteristics of sliding mode control will cause chattering of the system [9,10]. Research on SMC often focuses on weakening chattering as much as possible on the premise of ensuring the advantages of SMC. Adaptive SMC (ASMC) is proposed to avoid unnecessary chattering caused by the setting of switching gain of the sliding mode controller. This method makes the switching gain change adaptively by designing the switching gain adaptive law [11–13]. Super-twisted SMC (STSMC) theory makes the switching term more continuous and smoother by

transforming the switching term [14]. Similarly, the quasi-sliding mode method replaces the sign function in the traditional SMC by using the saturation function or continuous relay characteristics, thus reducing the discontinuity of switching terms and weakening chattering [15]. In addition, the CSMC method is proposed, which can not only weaken chattering but also has good tracking accuracy. Theoretically, it can be proven that the CSMC method has a smaller steady-state error bound than the quasi-sliding mode method with the saturation function [10,16,17]. However, the sliding mode controller needs to continuously increase the value of switching gain to maintain stability when there is a large amount of uncertainty inside and outside the system, which exacerbates control discontinuity. This makes the effectiveness of the improved sliding mode control method in weakening chattering less significant. Especially for some power electronic devices with high-frequency switches, excessive system vibration can reduce their service life and even cause safety issues [18,19].

Therefore, acquiring unknown uncertainties and compensating for them in the sliding mode controller can fundamentally weaken chattering without damaging the performance of the controller. Fortunately, neural networks (NNs) are widely used in the field of system identification and function approximation because they can approximate any continuous nonlinear function [20–24]. Therefore, in past studies, various networks have been proposed to obtain accurate and fast function approximation. Among them, RBFNNs are widely used in the field of function approximation and system identification with a simple structure and faster convergence speed than multi-layer perceptual networks [25,26]. Moreover, in order to improve the performance of RBFNNs, some improved RBFNN methods have been studied [13]. These networks add connections between hidden layer nodes and recursive structures between output layer nodes and hidden layer nodes. Without exception, these optimized RBFNNs greatly increase the complexity of the control algorithm, especially by increasing the number of hidden layers and adopting a recursive structure. In modern digital control systems, the realization of a control algorithm usually depends on the computing power of the control unit. With the increase in the complexity of the control algorithm, this will inevitably lead to the limited sampling frequency of the system and the aggravation of the algorithm delay [27]. For the above reasons, a simple network structure that can improve the approximation accuracy of NNs is needed. The functional link NN (FLNN), as a single-layer network based on extension polynomials, happens to meet this requirement. As a member of the FLNN, the Chebyshev network (CN) can also expand with nonlinear functions to generate nonlinear decision boundaries, thus forming complex decision regions. Meanwhile, functional expansion can effectively increase the dimension of the input vector, so that the hyperplane generated by the network will have good resolution in the input data space [28]. Moreover, the CN is widely used in the field of intelligent control due to its recursive characteristics and optimal approximation theory [29,30]. The existing research has proven that the unified model NN based on Chebyshev polynomials not only has function approximation ability but also has a faster learning speed than the traditional feedforward or recursive NNs [31,32]. The parameter learning methods of NNs and gradient descent and genetic algorithms are investigated in [33,34].

In addition, in order to avoid the problem of all hidden layer nodes being far away from the input signal, resulting in a network output of almost zero, this article proposes a structural self-evolution mechanism. Based on the distance between the center value vector of the Gaussian basis function and the input signal, it is determined whether it is necessary to add new nodes to expand the mapping range of the network. In summary, a self-evolving Chebyshev radial basis function neural network (SECRBFNN) is proposed to remove the influence of unknown nonlinear impact. The initial hidden layer node of the proposed SECRBFNN is one, and the number of final nodes and new node parameters are determined by the actual control situation. Therefore, the integration of this mechanism ensures that the structure of the proposed network can automatically evolve according to the actual situation when facing different application scenarios. Eventually, a shunt active

power filter (SAPF) is used as a simulation model to verify the designed control method. The main research contributions are summarized as:

- (1) The designed new CSMC algorithm is oriented to a multi-dimensional uncertain nonlinear system and does not depend on an accurate mathematical model. The superiority of the proposed method is explored to show that it has a smaller steady-state error.
- (2) A network combining the CN and the RBFNN is designed. The network combines the advantages of the CN and the RBFNN in function approximation and is not obvious in increasing the complexity of the algorithm. At the same time, the structure self-evolution mechanism is added to the network, so that the network structure can be adjusted according to the actual control scenario.
- (3) The SAPF model is adopted in the simulation phase. CSMC and the proposed method are respectively applied to the SAPF current control system. The comparative study between two methods is accomplished to highlight the superiority of the proposed method, from the perspectives of current tracking, harmonic compensation, and network operation time.

2. Problem Statement and Preliminaries

Consider the following N-order model with single input and single output:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dots \\ \dot{x}_n = f(X) + b(X)u \\ y = x_1 \end{cases} \tag{1}$$

where $X = [x_1 \ x_2 \ \dots \ x_n]^T$, X is the system state, and y is an output of the system. $f(X)$ and $b(X)$ are the nonlinear dynamics and control gains which satisfy the following assumptions:

Assumption 1. $f(X)$ and $b(X)$ are all bounded and satisfy:

$$|f(X)| < F(X) \tag{2}$$

$$|b(X)| < B(X) \tag{3}$$

where $F(X)$ and $B(X)$ are the positive bounded function.

Remark 1. In practical applications, objective factors such as equipment aging, temperature, humidity, etc., are highly likely to cause the actual model parameters to differ from the nominal values. In addition, human modeling errors and measurement errors are also important reasons for the occurrence of unknown uncertainties. Therefore, designing controllers using nominal values is bound to be influenced by unknown uncertainties. In this article, unknown uncertainty is summarized as a term in system dynamics [35,36].

Therefore, (1) can be rewritten as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dots \\ \dot{x}_n = f_0(X) + b_0(X)u + \Delta \\ y = x_1 \end{cases} \tag{4}$$

where $f_0(x)$ and $b_0(x)$ are the nonlinear dynamics and control gains with nominal value, respectively, and Δ is the lumped unknown nonlinear impact, which is assumed to be Lipschitz continuous and bounded and satisfies:

$$|\Delta| < D \tag{5}$$

where D is a positive constant.

3. Proposed Control Method and Stability Verification

The block diagram of the proposed control method is shown in Figure 1, where the SECRBFNN is used to obtain $\hat{\Delta}$ and CSMC to form the new ICSMC. Meanwhile, parameter learning laws and structural self-evolution mechanisms are used to adjust the parameters and structure of the network based on the actual control situations. In the Figure 1, e represents the tracking error, which can be written as:

$$e = y - y_d \tag{6}$$

where y_d is a desired signal. The control goal is to design a controller with good performance so that y can track y_d accurately and robustly.

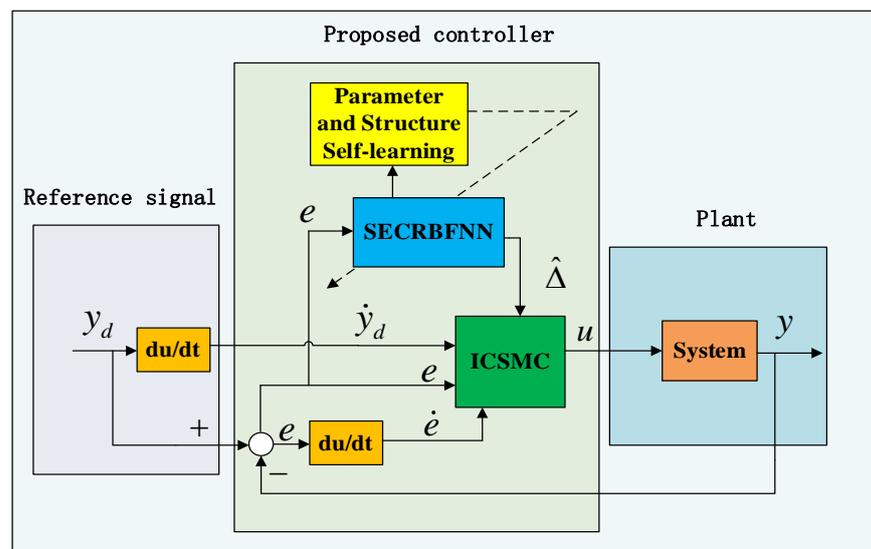


Figure 1. Block diagram of the proposed control systems.

Thus, \dot{y}_d and \dot{e} are the first derivative of expected signal in control system and tracking error, respectively; u represents control law.

3.1. Structure of the Self-Evolving Chebyshev Radial Basis Function Neural Network

The proposed SECRBFNN's structure is described in Figure 2, which is a four-layer network. Firstly, the input data are processed using Chebyshev extended polynomials, and then function approximation is achieved through the RBFNN. The following is an expression of the relationship between the input signals and output signals of each layer.

- (1) Layer 1—the first input layer: This layer is used to transfer the input signal. For each node, the relationship between the output and input are as follows:

$$y_i^{(1)} = u_i^{(1)}, (i = 1, 2, \dots, m) \tag{7}$$

$$u_i^{(1)} = x_i \tag{8}$$

where $y_i^{(1)}$ and $u_i^{(1)}$ are the output and input signal of the i -th node, respectively. The input of the entire network is the tracking error.

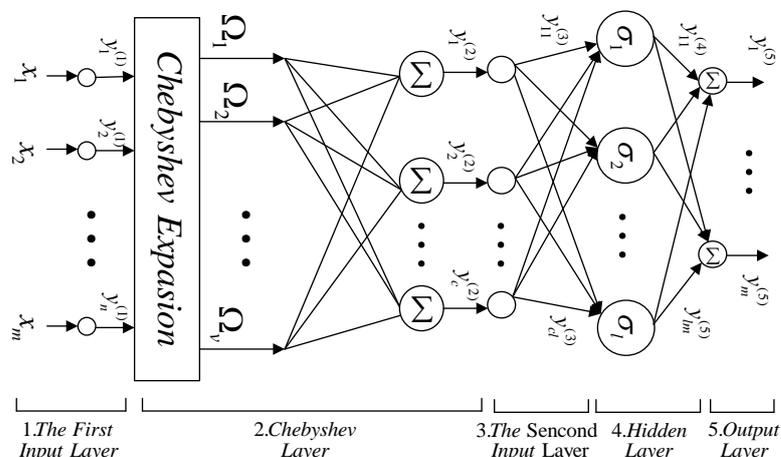


Figure 2. The structure of the SECRBFNN.

- (2) Layer 2—Chebyshev layer: Chebyshev polynomials are used to map the NN input to a higher dimensional space. In the high-dimensional space, the input signals can be divided as:

$$\begin{aligned} \Omega &= [\Omega_1, \Omega_2, \dots, \Omega_v]^T \\ &= [C_0, C_1(y_1^{(1)}), \dots, C_p(y_1^{(1)}), C_1(y_2^{(1)}), \dots, C_p(y_2^{(1)}), \\ &\dots, C_1(y_m^{(1)}), \dots, C_p(y_m^{(1)})]^T \end{aligned} \tag{9}$$

where Ω is the output vector after Chebyshev expansion, $C_h(x_i)$ is the Chebyshev polynomials, and $h = 1, 2, \dots, p$ represents the order of a polynomial. Chebyshev polynomials of each order are expressed as:

$$\begin{aligned} C_0 &= 1; \\ C_1(x_i) &= x_i \\ C_2(x_i) &= 2x_i^2 - 1 \\ C_3(x_i) &= 4x_i^3 - 3x_i \\ &\vdots \\ C_{h+1}(x_i) &= 2x_i C_h(x_i) - C_{h-1}(x_i) \end{aligned} \tag{10}$$

Finally, the output of this layer can be written as:

$$y_k^{(2)} = \sum_{r=1}^v o_{rk}^{(2)} \Omega_r, \quad k = 1, 2, \dots, c; \quad r = 1, 2, \dots, v \tag{11}$$

where $y_k^{(2)}$ is the output signal of the k -th output node of this layer and $o_{rk}^{(2)}$ represents the corresponding combined weight of the Chebyshev basis functions.

- (3) Layer 3—the second input layer: This layer is the bridge between the CNN and the RBFNN. The output is expressed as:

$$y_{kj}^{(3)} = u_{kj}^{(3)}, \quad (j = 1, 2, \dots, l) \tag{12}$$

$$u_{kj}^{(3)} = y_k^{(2)} \tag{13}$$

where $y_{kj}^{(3)}$ is the output signal of this layer from the k -th output node in the Chebyshev layer, which is sent to the j -th node in the hidden layer.

- (4) Layer 4—hidden layer: The mapping of input signals in this layer is the key to the performance of the RBFNN [35]. The output of this layer can be expressed as:

$$y_j^{(4)} = e^{-net_j^4}, \quad net_j^4 = \sum_{c=1}^k \frac{(u_{kj}^{(4)} - m_{kj})^2}{(\delta_{kj})^2} \tag{14}$$

$$u_{kj}^{(4)} = y_{kj}^{(3)} \tag{15}$$

where m_{kj} and δ_{kj} are the center and width in the Gaussian function, calculated in the j -th node for the k -th input, respectively. $y_j^{(4)}$ represents the output of the j -th hidden layer node.

- (5) Layer 5—output layer: This layer is used to transport the output signal of the entire network. The expression of the output of this layer is as follows:

$$y_o^{(5)} = \sum_{j=1}^l w_{jo}^{(5)} u_{jo}^{(5)}, \quad o = 1, 2, \dots, N \tag{16}$$

$$u_{jo}^{(5)} = y_j^{(4)} \tag{17}$$

where w_{jo} is the output weight between the j -th hidden layer node and the o -th output layer node; $y_o^{(5)}$ is the output of the o -th output layer node; and u_{jo} is the input of the o -th output layer node from the o -th output layer node.

We summarize all self-learning parameter vectors as follows:

$$W = [W_1, W_2, \dots, W_n] \in \mathbb{R}^n \tag{18}$$

$$M = [m_{11} \dots m_{c1}, m_{12} \dots m_{c2}, \dots, m_{1l} \dots m_{cl}]^T \in \mathbb{R}^{c \cdot l} \tag{19}$$

$$\delta = [\delta_{11} \dots \delta_{c1}, \delta_{12} \dots \delta_{c2}, \dots, \delta_{1l} \dots \delta_{cl}]^T \in \mathbb{R}^{c \cdot l} \tag{20}$$

3.2. Self-Evolving Mechanism

RBFNNs have been widely used in the field of function approximation, but a thorny problem is that they may not have good approximation ability due to the inability of all hidden layer nodes to map the current input signal well. Therefore, to avoid this problem, setting up multiple nodes is a common approach. However, in practical applications, multiple network nodes can lead to an increase in algorithm complexity, thereby increasing the cost of algorithm implementation. Therefore, setting an appropriate number of nodes has become the primary issue in neural network applications. This section proposes a structural self-evolution mechanism. At 0 s, the number of hidden layer nodes in the SECRBFNN is set to 1, and then, nodes are added when the activation intensity of the hidden layer does not meet the preset value. The initial value of the parameters of the new node is determined by the current input, ensuring that at least one hidden layer node can map the current input signal well.

The initial number of nodes in the hidden layer is set to 1. Equation (21) is used to calculate the generalized distance matrix between the input signal and the hidden layer node. The generalized distance matrix reflects the degree of correlation between the current input and the hidden layer nodes at the previous time. The result is proportional to the difference between the input signal and the center value of the previous moment $m_j(t - 1)$

and inversely proportional to the base width. The influence of the base width and center value on the mapping ability of Gaussian counting functions is explained in Remark 2:

$$d_j = (u_j(t) - m_j(t - 1))^T \Lambda_j^{-2} (u_j(t) - m_j(t - 1)) \tag{21}$$

where d_j is the distance matrix between the current input $u_j(t)$ and the existing j -th hidden layer nodes; the input signal of the j th hidden layer node in the fourth layer of the network at current time $u_j(t) = [u_{1j}^{(4)} \ u_{2j}^{(4)} \ \dots \ u_{kj}^{(4)}]^T$; and $m_j(t - 1) = [m_{1j} \ m_{2j} \ \dots \ m_{kj}]^T$ is the center value vector of the j -th hidden layer node at the previous moment and $\Lambda_j = \text{diag}(\delta_{1j}(t - 1), \delta_{2j}(t - 1) \dots, \delta_{kj}(t - 1))$.

Finding the hidden layer node closest to the current input:

$$d_{\min} = \min_{j=1,2,\dots,N(t-1)} d_j \tag{22}$$

If

$$d_{\min} > d_{th} \tag{23}$$

which indicates that the current effective mapping range of the hidden layer nodes does not include the input signal at this moment. Therefore, a new hidden layer node needs to be customized according to the current input. The initial parameters of the new node are:

$$m_{new} = x(t) \tag{24}$$

$$\delta_{new} = \delta_{init} \tag{25}$$

$$w_{new} = 0 \tag{26}$$

Remark 2. The center value and width value of the Gaussian function determine the mapping ability of the hidden layer nodes to specific input signals. The closer the center value is to the input signal, the more sensitive the Gaussian function is. Therefore, the selection of initial values for the newly added node center value vector in this article depends on the current input signal vector. In NN identification applications, the initial value of the width is often set to a larger value to increase the coverage range of the Gaussian basis function and enhance the adaptability of the hidden layer to various input signals. After adding a node, the NN will enter the parameter self-learning process. Then, with parameter learning, d_{\min} will be recalculated, and it will be checked as to whether it meets the requirements. If d_{\min} meets the preset requirements, the new node is retained, otherwise the candidate node will be discarded.

3.3. Controller Design and Stability Proof

N-dimensional generalized and complementary sliding surfaces are designed, respectively, as follows:

$$S_g = (p + \lambda)^n \int_{t_0}^t e(\tau) d\tau \tag{27}$$

$$S_c = (p + \lambda)^{n-1} (p - \lambda) \int_{t_0}^t e(\tau) d\tau \tag{28}$$

where λ is a sliding mode parameter and p is a differential operator.

Thus, a key equation can be obtained:

$$\dot{S}_c + \lambda(S_g + S_c) = \dot{S}_g \tag{29}$$

The intelligent complementary sliding mode controller is designed as:

$$u(t) = u_{eq} + u_{sw} \tag{30}$$

$$u_{eq} = -\frac{1}{b_0} [f_0(X) + \hat{\Delta} - y_d^{(n)} + \sum_{k=0}^{n-1} \binom{n+1}{k+1} e^{(k)} \lambda^{n-k} + \lambda^{n+1} \int_{t_0}^t e(\tau) d(\tau)] \tag{31}$$

$$u_{sw} = -k_w \text{sat} \left(\frac{S_g + S_c}{\phi} \right) \tag{32}$$

where u_{eq} represents an equivalent control law and u_{sw} is a switching control law, and $\text{sat}(\cdot)$ is the saturation function, which can be represent as:

$$\text{sat} \left(\frac{S_g + S_c}{\phi} \right) = \begin{cases} \text{sgn}(S_g + S_c) & |S_g + S_c| \geq \phi \\ \frac{S_g + S_c}{\phi} & |S_g + S_c| < \phi \end{cases} \tag{33}$$

where ϕ is the boundary-layer thickness.

According to the approximation theory of NNs, Δ and the approximation of $\hat{\Delta}$ can be expressed as:

$$\Delta = W^{*T} \sigma^*(M^*, \delta^*, x) + \varepsilon(x) \tag{34}$$

$$\hat{\Delta} = \hat{W}^T \hat{\sigma}(\hat{M}, \hat{\delta}, x) \tag{35}$$

where W^* and $\sigma^*(M^*, \delta^*, x)$ are the optimal output weight and the output of the hidden layer, respectively; M^* and δ^* are the optimal center value and width of the hidden layer, respectively; and $\varepsilon(x)$ represents the smallest approximation error. Similarly, \hat{W} and $\hat{\sigma}(\hat{M}, \hat{\delta}, x)$ are the actual output weight and the output of the hidden layer, respectively, and $\hat{M}, \hat{\delta}$ are the actual center value and width of the hidden layer.

Therefore, the approximation error in the NN can be deduced as:

$$\begin{aligned} \Delta - \hat{\Delta} &= W^{*T} \sigma^* - \hat{W}^T \hat{\sigma} + \varepsilon(x) \\ &= W^{*T} (\hat{\sigma} + \tilde{\sigma}) - \hat{W}^T \hat{\sigma} + \varepsilon(x) \\ &= W^{*T} \hat{\sigma} + W^{*T} \tilde{\sigma} - \hat{W}^T \hat{\sigma} + \varepsilon(x) \\ &= \tilde{W}^T \hat{\sigma} + \hat{W}^T \tilde{\sigma} + \tilde{W}^T \tilde{\sigma} + \varepsilon(x) \end{aligned} \tag{36}$$

where $\tilde{\sigma} = \sigma^* - \hat{\sigma}$, $\tilde{W} = W^* - \hat{W}$.

The Taylor expansion of δ^* at $M^* = \hat{M}$ and $\delta^* = \hat{\delta}$; then, $\tilde{\sigma}$ can be rewritten as:

$$\begin{aligned} \tilde{\sigma} &= \left. \frac{\partial \tilde{\sigma}}{\partial M} \right|_{M=\hat{M}} (M^* - \hat{M}) + \left. \frac{\partial \tilde{\sigma}}{\partial \delta} \right|_{\delta=\hat{\delta}} (\delta^* - \hat{\delta}) + O_h \\ &= D\sigma(M) \cdot \tilde{M} + D\sigma(\delta) \cdot \tilde{\delta} + O_h \end{aligned} \tag{37}$$

where O_h represents high-order terms in Taylor expansion expressions, and

$$D\sigma(M) = \begin{bmatrix} \frac{\partial \sigma_1}{\partial m_{11}} & \cdots & \frac{\partial \sigma_1}{\partial m_{cl}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \sigma_l}{\partial m_{11}} & \cdots & \frac{\partial \sigma_l}{\partial m_{cl}} \end{bmatrix}_{l \times cl} \tag{38}$$

$$D\sigma(\delta) = \begin{bmatrix} \frac{\partial \sigma_1}{\partial \delta_{11}} & \cdots & \frac{\partial \sigma_1}{\partial \delta_{cl}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \sigma_l}{\partial \delta_{11}} & \cdots & \frac{\partial \sigma_l}{\partial \delta_{cl}} \end{bmatrix}_{l \times cl} \tag{39}$$

Therefore,

$$\begin{aligned} \Delta - \hat{\Delta} &= \tilde{W}^T \hat{\sigma} + \hat{W}^T \tilde{\sigma} + \tilde{W}^T \tilde{\sigma} + \varepsilon(x) \\ &= \tilde{W}^T \hat{\sigma} + \hat{W}^T [D\sigma(M) \cdot \tilde{M} + D\sigma(\delta) \cdot \tilde{\delta} + O_h] + \tilde{W}^T \tilde{\sigma} + \varepsilon(x) \\ &= \tilde{W}^T \hat{\sigma} + \hat{W} D\sigma(M) \cdot \tilde{M} + \hat{W} D\sigma(\delta) \cdot \tilde{\delta} + e_{\Delta} \end{aligned} \tag{40}$$

where $e_{\Delta} = \hat{W}^T O_h + \tilde{W}^T \tilde{\sigma} + \varepsilon(x)$ is the lumped high-order approximation error and assuming $|e_{\Delta}| \leq \bar{e}_{\Delta}$.

The parameter learning methods of NNs include gradient descent, genetic algorithms [36], and other methods. In order to make the proposed network achieve the optimal approximation in theory, the parameter learning law is designed as follows:

$$\dot{\tilde{W}} = -\eta_1 (S_g + S_c) \hat{\sigma} \tag{41}$$

$$\dot{\tilde{\delta}}^T = -\eta_2 (S_g + S_c) \hat{W}^T D\sigma(\delta) \tag{42}$$

$$\dot{\tilde{M}}^T = -\eta_3 (S_g + S_c) \hat{W}^T D\sigma(M) \tag{43}$$

where η_1, η_2 , and η_3 are the parameter learning rates and all are positive constants.

Considering the following Lyapunov function:

$$V = \frac{1}{2} (S_g^2 + S_c^2) + \frac{1}{2\eta_1} \tilde{W}^T \tilde{W} + \frac{1}{2\eta_2} \tilde{\delta}^T \tilde{\delta} + \frac{1}{2\eta_3} \tilde{M}^T \tilde{M} \tag{44}$$

Taking the time derivative of (44) and using (30)–(33), we obtain:

$$\begin{aligned} \dot{V} &= S_g \dot{S}_g + S_c \dot{S}_c + H \\ &= (S_g + S_c) (\dot{S}_g - \lambda S_c) + H \\ &= (S_g + S_c) \left\{ p[(p + \lambda)^n \int_{t_0}^t e(\tau) d\tau] - \lambda S_c \right\} + H \\ &= (S_g + S_c) \left\{ e^{(n)} - \lambda S_c + [p(p + \lambda)^n \int_{t_0}^t e(\tau) d\tau - e^{(n)}] \right\} + H \\ &= (S_g + S_c) \left\{ x^{(n)} - x_d^{(n)} - \lambda S_c + [p(p + \lambda)^n \int_{t_0}^t e(\tau) d\tau - e^{(n)}] \right\} \\ &= (S_g + S_c) \left\{ f(x) + bu + \Delta - x_d^{(n)} - \lambda S_c + [p(p + \lambda)^n \int_{t_0}^t e(\tau) d\tau - e^{(n)}] \right\} + H \\ &= (S_g + S_c) \left\{ \Delta - \hat{\Delta} - \sum_{k=0}^{n-1} \binom{n+1}{k+1} e^{(k)} \lambda^{n-k} - \lambda^{n+1} \int_{t_0}^t e(\tau) d\tau \right. \\ &\quad \left. - \lambda S_c + [p(p + \lambda)^n \int_{t_0}^t e(\tau) d\tau - e^{(n)}] \right\} + H \\ &= (S_g + S_c) \left\{ \Delta - \hat{\Delta} - \lambda S_g - \lambda S_c - k_w \text{sat}\left(\frac{S_g + S_c}{\phi}\right) \right\} + H \end{aligned} \tag{45}$$

where $H = \frac{1}{\eta_1} \tilde{W}^T \dot{\tilde{W}} + \frac{1}{\eta_2} \tilde{\delta}^T \dot{\tilde{\delta}} + \frac{1}{\eta_3} \tilde{M}^T \dot{\tilde{M}}$.

By substituting parameter learning laws and (40), we can obtain:

$$\begin{aligned} \dot{V} &= -\lambda (S_g + S_c)^2 + [e_{\Delta} - k_w \text{sat}\left(\frac{S_g + S_c}{\phi}\right)] (S_g + S_c) \\ &\leq -\lambda (S_g + S_c)^2 + \bar{e}_{\Delta} |S_g + S_c| - k_w \text{sat}\left(\frac{S_g + S_c}{\phi}\right) (S_g + S_c) \end{aligned} \tag{46}$$

Thus, if $k_w > \bar{e}_{\Delta}$ is satisfied when $|S_g + S_c| \geq \phi$, the Lyapunov function is negative definite. This ensures that any position error trajectory will reach the boundary layer $|S_g + S_c| < \phi$ in finite time [17]. Stability proof of the proposed controller is completed.

4. Simulation Study

4.1. Introduction of the Model and Parameters

To verify the feasibility of the proposed controller, a shunt active power filter (SAPF) was used as the control object. The SAPF’s control system can be divided into three main parts: power supply voltage, nonlinear load, and the SAPF’s main circuit. The principle of a SAPF for harmonic suppression can be simply summarized as enabling the output current of the SAPF, namely the compensation current, to track the expected current calculated by the harmonic detection algorithm. Then, the compensating current is reversely injected into the power grid to counteract the harmonic current caused by nonlinear loads.

The circuit topology diagram of the SAPF is shown in Figure 3. The simulation model parameters are shown in Table 1. In order to simulate the system uncertainties caused by aging in actual circuits, in the MATLAB/Simulink simulation model, the inductance L and resistance R values in the SAPF main circuit are changed to 18 mH and 1 Ω , respectively. However, the nominal values in Table 1 were used in the controller design.

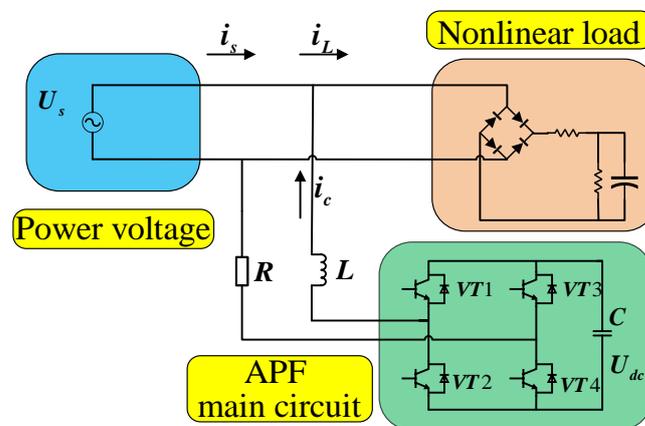


Figure 3. Circuit topology diagram of the SAPF.

Table 1. System parameters for simulation.

Parameters	Value
Grid voltage and frequency	24 V/50 Hz
Nonlinear load at steady state	$R = 5 \Omega, R_2 = 15 \Omega, C = 10^{-3} \text{ F}$
Additional nonlinear loads in parallel	$R = 15 \Omega, R_2 = 15 \Omega, C = 10^{-3} \text{ F}$
Main circuit parameters of the SAPF	$L = 10^{-3} \text{ H}, R = 0.1 \Omega, U_{dc} = 50 \text{ V}$
Sampling period	$T_s = 10^{-5} \text{ s}$

For PWM-based models, the most common modeling method is state space averaging, which converts a nonlinear time-varying switching circuit into an equivalent nonlinear time-varying continuous circuit mainly by weighted averaging of state variables. Therefore, the average dynamic model of an active filter can be written as:

$$\begin{aligned} \frac{di_c}{dt} &= (1 - u) \left(-\frac{R}{L} i_c - \frac{U_s + U_{dc}}{L} \right) + u \left(-\frac{R}{L} i_c + \frac{U_{dc} - U_s}{L} \right) \\ &= -\frac{R}{L} i_c - \frac{U_{dc} + U_s}{L} + \frac{2U_{dc}}{L} u \end{aligned} \tag{47}$$

where i_c represents the compensation current, U_s is the power supply voltage, i_s is the power supply current, U_{dc} is the voltage between capacitor C ; L and R are the circuit inductance and resistance, respectively; and u is the duty cycle of the PWM control input, which is determined by four IGBTs.

Based on the above analysis, the model of the SAPF is written as:

$$\dot{x} = f(x) + bu + \Delta(x) \tag{48}$$

where x represents i_c , $f(x) = -\frac{R}{L}x - \frac{U_S+U_{dc}}{L}$, $b = \frac{2U_{dc}}{L}$ and $\Delta(x)$ represents unknown uncertainty.

Define current tracking error $e = i_c - i_r$, where i_r is a reference current. Consequently, according to the results deduced in Section 3, the control law for the SAPF current loop control is:

$$u_{eq} = -\frac{1}{b}[f(x) + \hat{\Delta} - \ddot{x}_d + \lambda^2 e + 2\lambda \dot{e} + \lambda S_g] \tag{49}$$

$$u_{sw} = -K_w \text{sat}(S_g + S_c) \tag{50}$$

$$u = u_{eq} + u_{sw} \tag{51}$$

In order to highlight the superiority of the proposed method, traditional CSMC is used as a comparative method. The parameters of both methods are shown in Table 2. In addition, the parameters of DC side voltage control are also displayed.

Table 2. Parameters of the controller in the simulation.

Method	Parameters
CSMC	$k_w = 0.7, \lambda = 30$
New CSMC	$k_w = 0.6, \lambda = 30, d_{th} = 800, \eta_w = 3 \times 10^6, \eta_m = 0.005,$ $\eta_\delta = 0.03, \phi = 0.05$
Voltage loop PI controller	$K_p = 0.15, K_I = 0.02$

4.2. Steady-State Study

During steady-state response, the load in the power grid remains constant. Figure 4 shows the load current, compensation current, and power supply current curve (from top to bottom) under steady-state response. Not surprisingly, the current flowing through a nonlinear load is severely distorted. When the compensation current precisely tracks the reference current detected by the harmonic detection algorithm and is injected into the grid in the reverse direction, the power supply current changes to a standard sinusoidal signal.

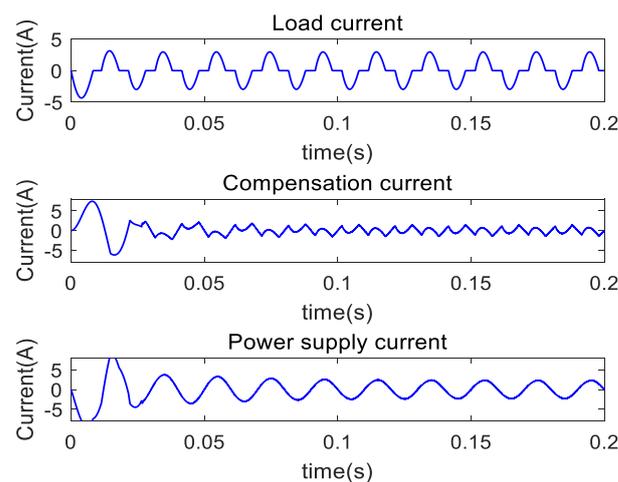


Figure 4. Load current, compensation current, and power supply current (from top to bottom).

Figure 5 shows the situation of compensating for current tracking of the reference current in SAPF current loop control. Within 0.03 s, the compensation current accurately tracks the expected current and the tracking error amplitude is only about 0.05, which demonstrates that the proposed method has an accurate steady-state tracking ability.

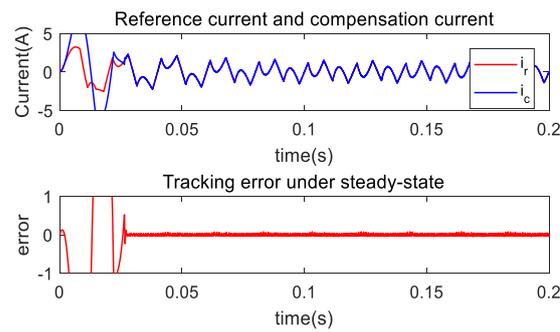


Figure 5. Current tracking and tracking error.

Figures 6 and 7 show the spectral analysis of the power supply current before and after SAPF connection. When the SAPF is not connected, the total harmonic distortion (THD) index of the grid current is as high as 36.81%, while after the SAPF is connected, the THD value of the power supply current is 1.58%. This indicates that most harmonics in the grid have been compensated, and the proposed method has good steady-state compensation ability.

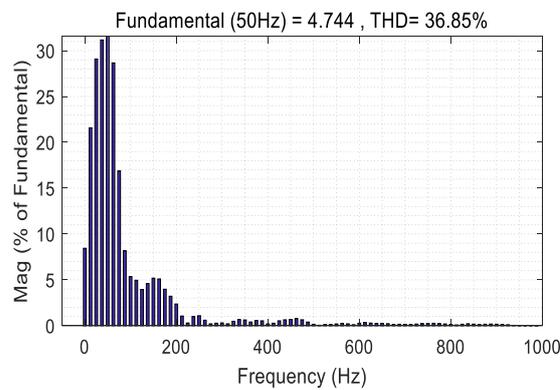


Figure 6. Power supply current spectrum without compensation.

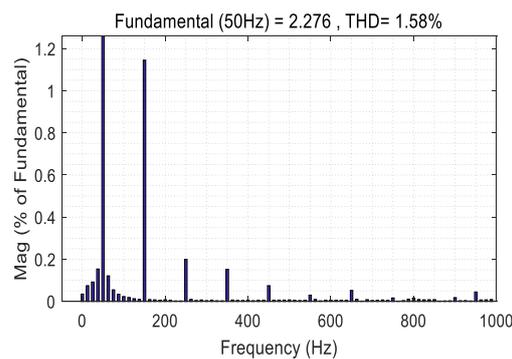


Figure 7. Power supply current spectrum after compensation.

4.3. Dynamic Study

The parallel connection and removal of loads in power networks are common phenomena in power networks. Therefore, in order to test whether the proposed method still maintains good performance under load variation, another nonlinear load will be parallel to the power grid at 0.3 s and removed at 0.6 s.

Figure 8 shows the load current, compensation current, and power supply current (from top to bottom) under dynamic response. Evidently, the amplitude of the load current and power supply current increase when a nonlinear load parallels to the power grid. However, the power supply current maintains a standard sinusoidal waveform and is smooth as it increases.

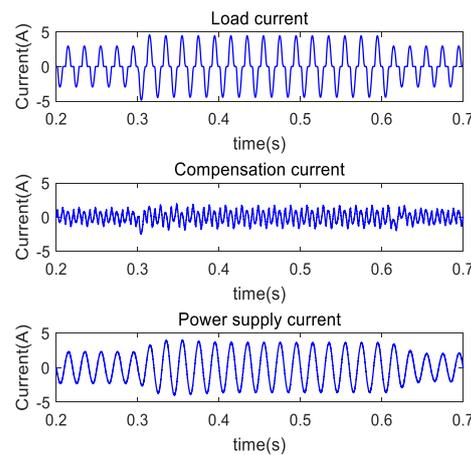


Figure 8. Load current, compensation current, and power supply current (from top to bottom) under dynamic response.

Figures 9 and 10 show the current tracing and tracking error when the load increases and decreases at 0.3 s and 0.6 s, respectively. Although the instantaneous tracking error increases instantaneously when the load increases at 0.3 s, the amplitude is only 0.1 and the duration is 0.01 s. This is due to the reference current changing at the moment of load change, which results in dynamic errors. However, since the tracking error is still small and the duration is short, the effect on the THD value in a cycle is not obvious.

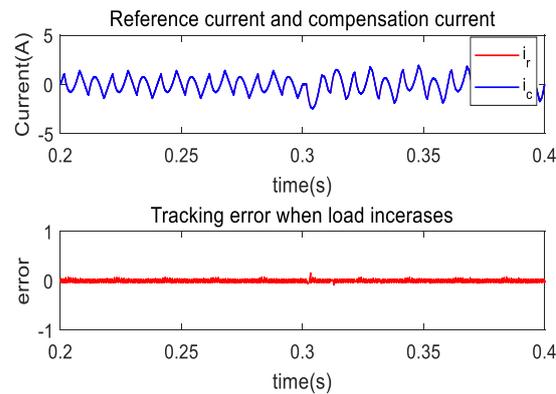


Figure 9. Current tracking and tracking error when the load increases.

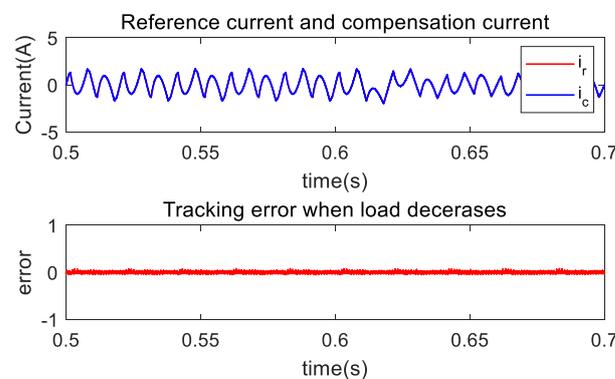


Figure 10. Current tracking and tracking error when the load decreases.

Figures 11 and 12 show the power grid current spectrum analysis after load increase and removal, respectively. Obviously, compared to the steady-state THD value of 1.58%, the THD value after load increase and removal is still very small at 1.22% and 1.89%,

respectively. The phenomenon reveals the fact that the performance of the proposed control method is not able to be influenced through load variation.

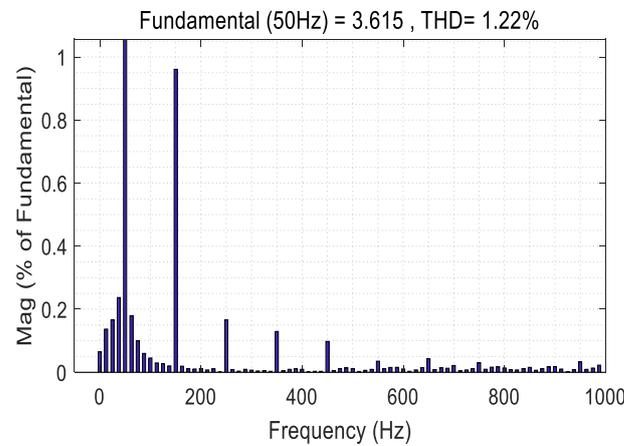


Figure 11. Power supply current spectrum after the load increases.

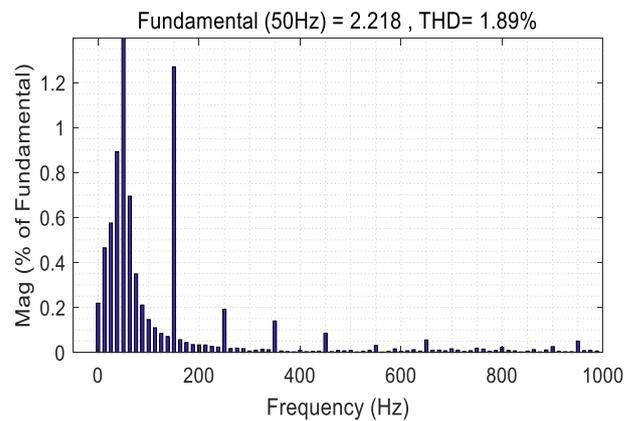


Figure 12. Power supply current spectrum after the load decreases.

Figure 13 describes the variation curve of the hidden layer nodes. Ultimately, the self evolution of the number of nodes to three can achieve the above control effect. Figure 14 shows the DC side voltage curve (blue curves). The SAPF voltage loop PI controller can maintain the voltage on both sides of the capacitor at around 50V (red dotted line), which also meets expectations.

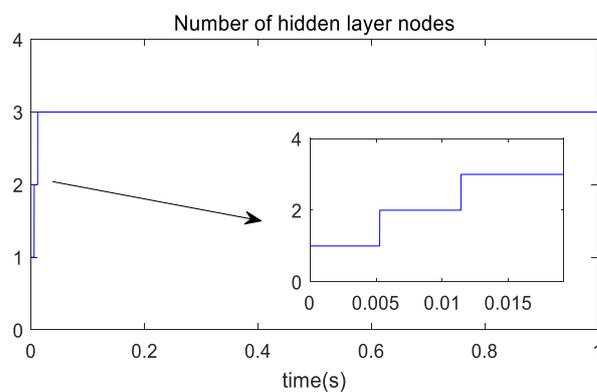


Figure 13. The variation curve of the number of hidden layer nodes.

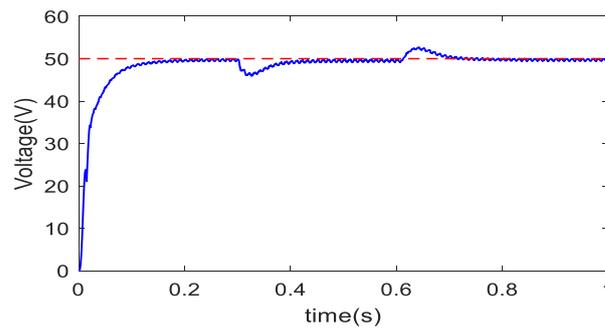


Figure 14. DC side voltage variation curve.

4.4. Comparison Study

In order to explore the superiority of the proposed method, CSMC is used as a comparative method in this part. Figure 15 shows the comparison of current tracking and tracking errors between the new CSMC and the CSMC.

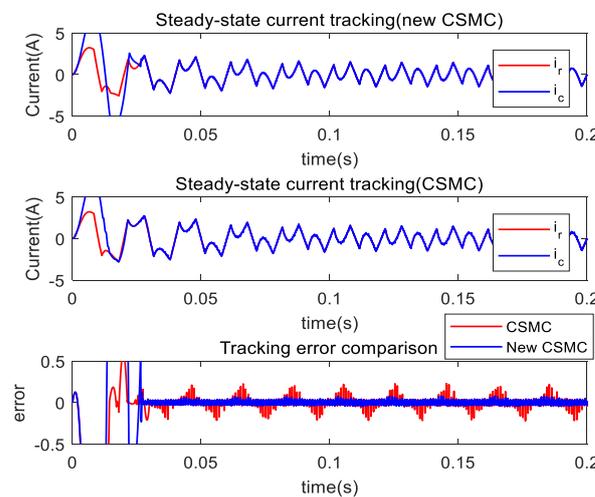


Figure 15. Tracking error comparison between the different methods.

As a control method insensitive to uncertainties such as external disturbances, parameter perturbations, and modeling errors, the CSMC method in the SMC family can also achieve accurate current tracking. However, its error waveform is not smooth and exhibits obvious periodicity, indicating that the suppression of the uncertainty of SAPF periodicity using CSMC is not sufficient. On the contrary, the current tracking error under new CSMC control is smoother, indicating that the unknown periodic uncertainty present in the system is compensated. If more outstanding control performance of CSMC is desired, a larger switching gain is needed. This will inevitably lead to an exacerbation of chattering. On the contrary, under the estimation of unknown uncertainty in the SAPF by the SECRBFNN, the switching gain of the CSMC part can be set to be smaller, and the chattering can be weakened as a result.

Table 3 shows the calculation time comparison of existing networks. Obviously, compared to the double-loop recurrent NN (DLRNN) [13] with two recursive structures and the double hidden layer recurrent NN (DHLRNN) [7] with multiple hidden layers, the proposed network has a shorter computational time.

Table 3. Computational time comparison.

Methods	RBFNN	DLRNN [12]	DHLRNN [7]	SECRBFNN
Time (s)	25.194	49.737	75.641	42.562

Remark 3. The NNs in Table 3 are all simulated in MATLAB/Simulink. The simulation method and simulation environment are identical. We used the built-in Profile Viewer function in MATLAB to calculate the running time of the four methods and used this as an indicator to evaluate whether the proposed method increases the computational burden while achieving good control effects.

Table 4 connects the THD index of the two methods under three states. The THD of the proposed method is lower than that of CSMC in both steady-state and load variation situations, indicating that the proposed method has better steady-state and dynamic harmonic compensation capabilities.

Table 4. THD index comparison.

State\Strategy	THD of CSMC	THD of the New CSMC
Steady-state	2.41%	1.58%
After the load increases	2.17%	1.22%
After the load decreases	2.64%	1.89%

5. Conclusions

This paper presents a new type of intelligent complementary sliding mode controller. Compared with the classical sliding mode controller, the proposed new CSMC uses a SECRBFNN to estimate the unknown uncertainties and to compensate for the effects of uncertainties. Since the influence of uncertainty in the system is compensated, the switching gain of the CSMC part can also be set to be smaller to reduce system chattering. The designed SECRBFNN combines parameter self-learning and structural self-evolution mechanisms by the combination of the CN and an RBFNN, enabling the structure and parameters of the SECRBFNN to be automatically adjusted. Eventually, the superiority of the proposed method was verified through simulation research on the SAPF model. The proposed method can achieve better current control performance than the classical CSMC method. In all three simulation scenarios, the THD of the grid current after APF connection controlled by the proposed method is lower than that of CSMC, and the network computational complexity is lower compared to the existing NN.

Author Contributions: Conceptualization, J.F.; methodology, L.Z.; writing—original draft preparation, L.Z.; writing—review and editing, J.F. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Science Foundation of China under grant No. 62273131.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

ICSMC	Intelligent complementary sliding mode control
RBFNN	Radial basis function neural network
SECRBFNN	Self-evolving Chebyshev radial basis function neural network
CSMC	Complementary sliding mode control
ASMC	Adaptive sliding mode control
STSMC	Super-twisted sliding mode control
CN	Chebyshev network
NN	Neural network
FLNN	Functional link NN

References

1. Fuhui, G.; Pingli, L. Fast self-adapting high-order sliding mode control for a class of uncertain nonlinear systems. *J. Syst. Eng. Electron.* **2021**, *32*, 690–699. [[CrossRef](#)]
2. Hong, Y.; Wang, J.; Cheng, D. Adaptive finite-time control of nonlinear systems with parametric uncertainty. *IEEE Trans. Autom. Control* **2006**, *51*, 858–862. [[CrossRef](#)]
3. Lin, F.-J.; Chen, C.-I.; Xiao, G.-D.; Chen, P.-R. Voltage Stabilization Control for Microgrid With Asymmetric Membership Function-Based Wavelet Petri Fuzzy Neural Network. *IEEE Trans. Smart Grid* **2021**, *12*, 3731–3741. [[CrossRef](#)]
4. Wang, A.; Liu, L.; Qiu, J.; Feng, G. Event-Triggered Robust Adaptive Fuzzy Control for a Class of Nonlinear Systems. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 1648–1658. [[CrossRef](#)]
5. Zhang, R.-Y.; Wu, L.-B.; Zhao, N.-N.; Yan, Y. Adaptive Event-Triggered Fuzzy Tracking Control of Uncertain Stochastic Nonlinear Systems With Unmeasurable States. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 2183–2196. [[CrossRef](#)]
6. Ang, K.H.; Chong, G.; Yun, L. PID control system analysis, design, and technology. *IEEE Trans. Control. Syst. Technol.* **2005**, *13*, 559–576. [[CrossRef](#)]
7. Fei, J.; Chen, Y. Dynamic Terminal Sliding-Mode Control for Single-Phase Active Power Filter Using New Feedback Recurrent Neural Network. *IEEE Trans. Power Electron.* **2020**, *35*, 9904–9922. [[CrossRef](#)]
8. Gwon, J.-S.; Lee, H.; Kim, S. Observer-Based FL-SMC Active Damping for Back-to-Back PWM Converter with LCL Grid Filter. *Int. J. Fuzzy Log. Intell. Syst.* **2015**, *15*, 200–207. [[CrossRef](#)]
9. Fei, J.; Chen, Y. Fuzzy Double Hidden Layer Recurrent Neural Terminal Sliding Mode Control of Single-Phase Active Power Filter. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3067–3081. [[CrossRef](#)]
10. Lin, F.-J.; Chou, P.-H.; Chen, C.-S.; Lin, Y.-S. DSP-Based Cross-Coupled Synchronous Control for Dual Linear Motors via Intelligent Complementary Sliding Mode Control. *IEEE Trans. Ind. Electron.* **2012**, *59*, 1061–1073. [[CrossRef](#)]
11. Liu, L.; Fei, J. Extended State Observer Based Interval Type-2 Fuzzy Neural Network Sliding Mode Control With Its Application in Active Power Filter. *IEEE Trans. Power Electron.* **2022**, *37*, 5138–5154. [[CrossRef](#)]
12. Baek, J.; Jin, M.; Han, S. A New Adaptive Sliding-Mode Control Scheme for Application to Robot Manipulators. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3628–3637. [[CrossRef](#)]
13. Fei, J.; Liu, L. Real-Time Nonlinear Model Predictive Control of Active Power Filter Using Self-Feedback Recurrent Fuzzy Neural Network Estimator. *IEEE Trans. Ind. Electron.* **2022**, *69*, 8366–8376. [[CrossRef](#)]
14. Balogoun, I.; Marx, S.; Liard, T.; Plestan, F. Super-Twisting Sliding Mode Control for the Stabilization of a Linear Hyperbolic System. *IEEE Control. Syst. Lett.* **2023**, *7*, 1–6. [[CrossRef](#)]
15. Janardhanan, S.; Bandyopadhyay, B. Multirate Output Feedback Based Robust Quasi-Sliding Mode Control of Discrete-Time Systems. *IEEE Trans. Autom. Control* **2007**, *52*, 499–503. [[CrossRef](#)]
16. Lin, F.-J.; Chen, S.-G.; Huang, M.-S.; Liang, C.-H.; Liao, C.-H. Adaptive Complementary Sliding Mode Control for Synchronous Reluctance Motor with Direct-Axis Current Control. *IEEE Trans. Ind. Electron.* **2022**, *69*, 141–150. [[CrossRef](#)]
17. Lin, F.-J.; Hwang, J.-C.; Chou, P.-H.; Hung, Y.-C. FPGA-Based Intelligent-Complementary Sliding-Mode Control for PMLSM Servo-Drive System. *IEEE Trans. Power Electron.* **2010**, *25*, 2573–2587. [[CrossRef](#)]
18. Fei, J.; Zhang, L.; Zhuo, J.; Fang, Y. Wavelet Fuzzy Neural Super-Twisting Sliding Mode Control of Active Power Filter. *IEEE Trans. Fuzzy Syst.* **2023**, 3272028. [[CrossRef](#)]
19. Zhang, L.; Fei, J. Intelligent Complementary Terminal Sliding Mode Using Multiloop Neural Network for Active Power Filter. *IEEE Trans. Power Electron.* **2023**, *38*, 9367–9383. [[CrossRef](#)]
20. Fei, J.; Feng, Z. Fractional-Order Finite-Time Super-Twisting Sliding Mode Control of Micro Gyroscope Based on Double-Loop Fuzzy Neural Network. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 7692–7706. [[CrossRef](#)]
21. Lin, F.-J.; Chen, S.-G.; Hsu, C.-W. Intelligent Backstepping Control Using Recurrent Feature Selection Fuzzy Neural Network for Synchronous Reluctance Motor Position Servo Drive System. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 413–427. [[CrossRef](#)]
22. Zhang, R.; Xu, B.; Shi, P. Output Feedback Control of Micromechanical Gyroscopes Using Neural Networks and Disturbance Observer. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 962–972. [[CrossRef](#)]
23. Fei, J.; Wang, H.; Fang, Y. Novel Neural Network Fractional-Order Sliding-Mode Control With Application to Active Power Filter. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 3508–3518. [[CrossRef](#)]
24. Fei, J.; Wang, Z.; Liang, X.; Feng, Z.; Xue, Y. Fractional Sliding-Mode Control for Microgyroscope Based on Multilayer Recurrent Fuzzy Neural Network. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 1712–1721. [[CrossRef](#)]
25. Lai, W.; Peter, W.T.; Zhang, G.; Shi, T. Classification of gear faults using cumulants and the radial basis function network. *Mech. Syst. Signal Process.* **2004**, *18*, 381–389. [[CrossRef](#)]
26. Zhang, L.; Li, K.; He, H.; Irwin, G.W. A new discrete-continuous algorithm for radial basis function networks construction. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 1785–1798. [[CrossRef](#)]
27. Stéphane, B.; Jin, H. Time delay compensation of digital control for DC switchmode power supplies using prediction techniques. *IEEE Trans. Power Electron.* **2000**, *15*, 835–842.
28. Chen, C.H.; Lin, C.J.; Lin, C.T. A Functional-Link-Based Neurofuzzy Network for Nonlinear System Control. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 1362–1378. [[CrossRef](#)]

29. Chen, S.-G.; Lin, F.-J.; Liang, C.-H.; Liao, C.-H. Intelligent Maximum Power Factor Searching Control Using Recurrent Chebyshev Fuzzy Neural Network Current Angle Controller for SynRM Drive System. *IEEE Trans. Power Electron.* **2021**, *36*, 3496–3511. [[CrossRef](#)]
30. Fei, J.; Wang, Z.; Fang, Y. Self-Evolving Recurrent Chebyshev Fuzzy Neural Sliding Mode Control for Active Power Filter. *IEEE Trans. Ind. Inform.* **2023**, *19*, 2729–2739. [[CrossRef](#)]
31. Chittora, P.; Singh, A.; Singh, M. Chebyshev Functional Expansion Based Artificial Neural Network Controller for Shunt Compensation. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3792–3800. [[CrossRef](#)]
32. Fei, J.; Wang, Z.; Pan, Q. Self-Constructing Fuzzy Neural Fractional-Order Sliding Mode Control of Active Power Filter. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 3169518. [[CrossRef](#)] [[PubMed](#)]
33. Vo, A.T.; Truong, T.N.; Kang, H.-J. Fixed-Time RBFNN-Based Prescribed Performance Control for Robot Manipulators: Achieving Global Convergence and Control Performance Improvement. *Mathematics* **2023**, *11*, 2307. [[CrossRef](#)]
34. Jäntschi, L.; Bolboacă, S.D.; Sestraș, R.E. Meta-heuristics on quantitative structure-activity relationships: Study on polychlorinated biphenyls. *J. Mol. Model.* **2010**, *16*, 77–86. [[CrossRef](#)] [[PubMed](#)]
35. Wang, J.; Fang, Y.; Fei, J. Adaptive Super-Twisting Sliding Mode Control of Active Power Filter Using Interval Type-2-Fuzzy Neural Networks. *Mathematics* **2023**, *11*, 2785. [[CrossRef](#)]
36. Zahraoui, Y.; Zaihidee, F.M.; Kermadi, M.; Mekhilef, S.; Mubin, M.; Tang, J.R.; Zaihidee, E.M. Fractional Order Sliding Mode Controller Based on Supervised Machine Learning Techniques for Speed Control of PMSM. *Mathematics* **2023**, *11*, 1457. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.