

Article

Adaptive Intelligent Sliding Mode Control of a Dynamic System with a Long Short-Term Memory Structure

Lunhaojie Liu ¹, Wen Fu ², Xingao Bian ² and Juntao Fei ^{1,2,3,*} ¹ College of IoT Engineering, Hohai University, Changzhou 213022, China; lunhaojieliu@hhu.edu.cn² College of Mechanical and Electrical Engineering, Hohai University, Changzhou 213022, China; 19921130@hhu.edu.cn (W.F.); 19881134@hhu.edu.cn (X.B.)³ Jiangsu Key Laboratory of Power Transmission and Distribution Equipment Technology, Hohai University, Changzhou 213022, China

* Correspondence: jtfei@hhu.edu.cn

Abstract: In this work, a novel fuzzy neural network (NFNN) with a long short-term memory (LSTM) structure was derived and an adaptive sliding mode controller, using NFNN (ASMC-NFNN), was developed for a class of nonlinear systems. Aimed at the unknown uncertainties in nonlinear systems, an NFNN was designed to estimate unknown uncertainties, which combined the advantages of fuzzy systems and neural networks, and also introduced a special LSTM recursive structure. The special three gating units in the LSTM structure enabled it to have selective forgetting and memory mechanisms, which could make full use of historical information, and have a stronger ability to learn and estimate unknown uncertainties than general recurrent neural networks. The Lyapunov stability rule guaranteed the parameter convergence of the neural network and system stability. Finally, research into a simulation of an active power filter system showed that the proposed new algorithm had better static and dynamic properties and robustness compared with a sliding controller that uses a recurrent fuzzy neural network (RFNN).

Keywords: fuzzy neural network; long short-term memory; adaptive sliding mode control**MSC:** 68T07; 93C40; 93C42

Citation: Liu, L.; Fu, W.; Bian, X.; Fei, J. Adaptive Intelligent Sliding Mode Control of a Dynamic System with a Long Short-Term Memory Structure. *Mathematics* **2022**, *10*, 1197. <https://doi.org/10.3390/math10071197>

Academic Editor: Moussa Labbadi

Received: 20 February 2022

Accepted: 4 April 2022

Published: 6 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

Most systems existing in nature are nonlinear. Many scholars have developed various linearization methods [1,2], which are then used for analysis and processing. However, especially in practical applications, for highly nonlinear systems, implementing control tasks for dynamic systems with uncertain parameters is still a hot research issue. In recent years, the scientific community has developed multiple advanced control strategies for nonlinear systems. SMC became the most widely used and effective control method in complex nonlinear systems [3–7].

However, traditional sliding mode control can often cause a system to have high-frequency switching characteristics, which may have a serious impact on the system. Recently, fuzzy logic systems (FLS) [8,9] and neural networks (NN) [10,11] have been widely applied in parameter estimation and system identification as two main schemes. Due to its fuzzy mechanism, FLS has a strong ability to deal with uncertain systems. In [8], a fuzzy logic control method was designed to change controller parameters in order to adapt the system uncertainty. Artificial neural networks (ANN) are adaptive and self-learning, and, theoretically, a multilayer feedforward NN can approximate any complex nonlinear function [9]. In [10], an NN controller was developed to approximate the upper bound of system uncertainties for nonlinear systems. Subsequently, various forms and structures of neural networks [11–15] were proposed for nonlinear control problems, such as the back propagation (BP) NN [12,13], radial basis function (RBF) NN [14,15], and so on. RBF neural networks have also been

developed and have changed rapidly. These neural networks can adaptively compensate for the nonlinearity in a system using an adaptive feedback controller, which is described in [16]. The learning algorithm of an RBF neural network was improved and a new generalized growth and pruning algorithm was proposed for RBF in [17]. An RBF neural network was applied to robot trajectory tracking control in [18]. A fuzzy neural network (FNN) that combined a fuzzy control method and an NN estimator were developed to approach unknown system uncertainties in [19,20]. A recurrent feature-selection-type FNN was employed for a synchronous reluctance motor in [21].

Considering that traditional BP neural networks and RBF neural networks are both feed-forward networks, it is not easy to handle the time correlation issue without using historical data. Therefore, for signals with complex features, various recurrent NN methods have been proposed [22–24]. Advanced NN strategies using sliding mode schemes have been researched for dynamic systems [25–27].

Hochreiter and Schmidhuber proposed a more effective long short-term memory (LSTM) NN structure in [28]. Some variants of LSTM were studied and the forgetting-gate property and output gate with output activation function were proved to be its most critical components [29]. The interpretability of an LSTM structure was studied in depth [30]. Due to its many compelling features, LSTM is used in areas such as motion recognition [31], video subtitles [32], image classification [33], nonlinear regression [34], and natural language processing [35]. LSTM was first applied to the field of control in [36], where LSTM was used to deal with the nonlinear dynamics and long-term time dependence that exists in human motion. Then, a new self-evolving interval type-2 fuzzy LSTMNN was proposed with the help of FLS and LSTM structures in [37]. Advanced intelligent control schemes were derived to approximate and provide a valid way of approaching for nonlinear systems [38,39].

In this paper, the LSTM mechanism was introduced into an FNN; hence, a novel fuzzy neural network (NFNN) with an LSTM structure was developed. Then, an adaptive sliding mode control method, using the NFNN, was derived for nonlinear systems. The main contributions of this work are discussed in the following:

- (1) Compared with existing work, this paper introduces the LSTM mechanism into the FNN and proposes a novel FNN sliding mode method. Using LSTM in this way could satisfactorily solve the issue of time dependence and vanishing gradients. There appears to be no need for parameter fine-tuning because an NFNN works for the learning rate and initial value of the weight.
- (2) Most systems have nonlinearity and unknown uncertainty, which bring many unforeseen problems to system control. In this study, an adaptive sliding mode method based on an NFNN was designed for a class of nonlinear systems. This method had the significant advantages of model-free control, with a wide range of applications, strong disturbance rejection ability, and good static and dynamic performance.
- (3) The robustness of the controller was the main performance index. In general, the faster and more accurate the estimation of the time-varying unknown uncertainty of the system, the better the robustness. The simulation studied the system performance in the presence of parameter changes of the APF circuit, and compared it with SMC and ASMC-RFNN, showing its stronger learning ability and robustness.

2. Problem Statement and Preliminaries

Consider a class of single-input single-output partially unknown nonlinear systems, which are shown by the differential equation in [10].

$$\begin{aligned} x^{(n)}(t) + f(x(t), \dot{x}(t), x^{(2)}(t), \dots, x^{(n-1)}(t)) = \\ b(x(t), \dot{x}(t), x^{(2)}(t), \dots, x^{(n-1)}(t))u(t) \end{aligned} \quad (1)$$

where t is time, $x^{(i)}(t)$ ($i = 2, \dots, n$) are the i -th time derivatives of the $x(t)$, $f(\cdot) : R^n \rightarrow R$, $b(\cdot) : R^n \rightarrow R$ are nonlinear functions, and $u(t) \in R$ is control input.

For general consideration, assumptions 1 and 2 are made.

Assumption 1. The nonlinear function of system $f(x(t), \dot{x}(t), x^2(t), \dots, x^{(n-1)}(t))$ is absolute value bounded:

$$|f(x(t), \dot{x}(t), x^2(t), \dots, x^{(n-1)}(t))| < f_b(x(t), \dot{x}(t), x^2(t), \dots, x^{(n-1)}(t)) \quad (2)$$

where $f_b(x(t), \dot{x}(t), x^2(t), \dots, x^{(n-1)}(t))$ is a positive function.

Assumption 2. The control gain $b(x(t), \dot{x}(t), x^2(t), \dots, x^{(n-1)}(t))$ is a known positive function, and it is lower bounded:

$$b(x(t), \dot{x}(t), x^2(t), \dots, x^{(n-1)}(t)) > b_l(x(t), \dot{x}(t), x^2(t), \dots, x^{(n-1)}(t)) \quad (3)$$

where $b_l(x(t), \dot{x}(t), x^2(t), \dots, x^{(n-1)}(t))$ is a positive function.

Then, considering parameter variances and disturbances, we can rewrite Equation (1) using state-space notation as follows:

$$\dot{X} = AX + Bu + F + \bar{\delta} \quad (4)$$

where $\bar{\delta} = [0, 0, \dots, \delta]^T \in R^{n \times 1}$ is the lumped uncertainty of the nonlinear system, which is caused by internal parameter perturbation and external disturbance; $X = [x, \dot{x}, x^{(2)} \dots, x^{(n-1)}]^T \in R^{n \times 1}$; $B = [0, \dots, b(X)]^T \in R^{n \times 1}$; $F = [0, \dots, -f(X)]^T \in R^{n \times 1}$;

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \in R^{n \times n}.$$

Control Objective: A controller was designed to ensure that X accurately tracked the reference trajectory X_m , so that the nonlinear system (4) was asymptotically stable, and all the signals were bounded.

To achieve the above objectives, a traditional sliding controller was designed for an n -order nonlinear system expressed by (4).

Denote a tracking error as:

$$E = X - X_m = [e, \dot{e}, \dots, e^{(n-1)}]^T \in R^{n \times 1} \quad (5)$$

where X_m is the n -order reference trajectory vector, and E is the n -order error vector.

The derivative of Equation (5) is:

$$\dot{E} = \dot{X} - \dot{X}_m \quad (6)$$

A standard sliding surface is designed as:

$$S = CE \quad (7)$$

where $C = [c_1, c_2, \dots, c_n] \in R^{1 \times n}$ is the parameter vector of the sliding surface.

Remark 1. In this paper, a conventional linear sliding mode surface was selected. Its advantage was that it was simple and practical. As long as the selected sliding mode surface parameters satisfied the Hurwitz condition, the asymptotic stability of the system could be guaranteed. Although the terminal sliding mode surface and nonsingular sliding mode surface developed in the follow-up research improved asymptotic stability to finite-time stability, they generated some nonsingularity-solving problems; hence, the design and implementation were slightly difficult. In future research,

the authors may employ some more advanced nonlinear sliding mode surfaces to reduce system chattering and achieve finite-time stabilization effects.

The derivative of Equation (7) is simplified by the Equations (4)–(7) and expressed as:

$$\begin{aligned} \dot{S} &= C\dot{E} = [c_1, c_2, \dots, c_n] \begin{bmatrix} \dot{e} \\ e^{(2)} \\ \vdots \\ e^{(n)} \end{bmatrix} \\ &= c_1\dot{e} + c_2e^{(2)} + \dots + c_{n-1}e^{(n-1)} + c_n(x^n - x_m^n) \\ &= c_1\dot{e} + c_2e^{(2)} + \dots + c_{n-1}e^{(n-1)} + c_n(b(X)u - f(X) + \delta - x_m^n) \end{aligned} \tag{8}$$

Letting $\dot{S} = 0$ to solve an equivalent control force u_{eq} :

$$\begin{aligned} u_{eq} &= \frac{1}{b(X)}(f(X) - \delta + x_m^n) - \frac{1}{c_nb(X)}(c_1\dot{e} \\ &\quad + c_2e^{(2)} + \dots + c_{n-1}e^{(n-1)}) \end{aligned} \tag{9}$$

Consequently, a new controller using a switching term is given as:

$$\begin{aligned} u &= \frac{1}{b(X)}(f(X) + x_m^n) - \frac{1}{c_nb(X)}(c_1\dot{e} + c_2e^{(2)} + \dots + c_{n-1}e^{(n-1)}) \\ &\quad - K\text{sgn}(S) \end{aligned} \tag{10}$$

where K a positive value, and $\text{sign}(\cdot)$ represents a sign function.

Remark 2. The sign function brought a certain amount of system chattering, so sat (saturation) or tanh functions were used instead to improve smoothness and reduce chattering. In addition, dynamic sliding mode and super-twisting sliding mode methods could also weaken the effect of chattering well, which could be investigated in further research work.

A Lyapunov function is chosen as follows:

$$V_1 = \frac{1}{2}S^T S \tag{11}$$

Then, the derivative of V_1 is derived as

$$\dot{V}_1 = S^T(c_1\dot{e} + c_2e^{(2)} + \dots + c_{n-1}e^{(n-1)} + c_n(b(X)u - f(X) + \delta - x_m^n)) \tag{12}$$

Substituting (10) into (12) the following is obtained:

$$\begin{aligned} \dot{V}_1 &= S^T(\delta - c_nb(X)K\text{sgn}(S)) \\ &= -c_nb(X)K|S| + S^T\delta \\ &\leq -c_nb_lK|S| + |S|\delta_b = |S|(\delta_b - c_nb_lK) \end{aligned} \tag{13}$$

where $b(X)$ is a known function with a positive lower bound b_l , and δ is a lumped uncertainty with upper bound δ_b . When $K \geq \delta_b/c_nb_l$, inequality (13) satisfies $\dot{V}_1 \leq 0$. From Lyapunov theory, the system is asymptotically stable.

3. The Novel FNN Structure

The unknown $f(X)$ needs to be used in the controller (10); therefore, in practical applications, the unknown $f(X)$ needs to be estimated. A novel FNN with an LSTM structure was adopted to identify the $f(X)$, as shown in Figures 1 and 2. Its unique

recursive structure with LSTM greatly increased the ability to approximate unknown time-varying functions.

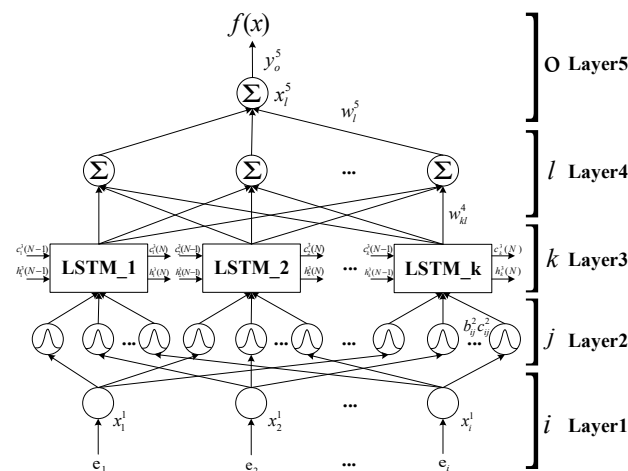


Figure 1. Block diagram of the NFNN.

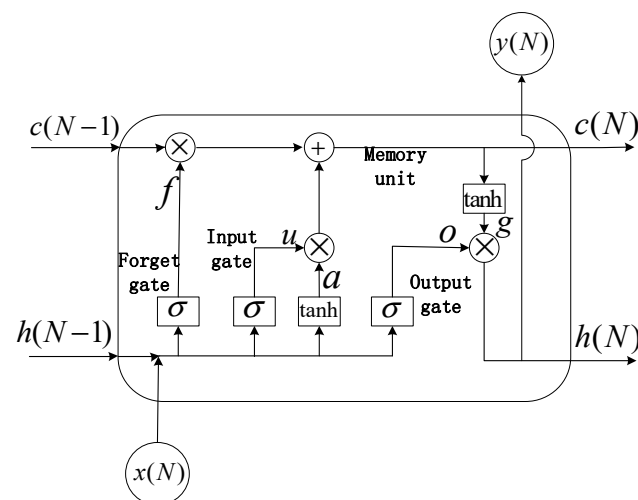


Figure 2. Structure of LSTM.

The basic rules and signal transmission of each layer of an NFNN are given in the following steps.

Layer 1—Input Layer: For node i of the input layer, the node input and output were derived as:

$$net_i^1(N) = x_i^1(N) \tag{14}$$

$$y_i^1(N) = f_i^1(net_i^1(N)) = net_i^1(N), i = 1 \tag{15}$$

where superscript and subscript show the number of layer nodes respectively; $x_i^1(N)$ is the input, $net_i^1(N)$ are the network inputs, $y_i^1(N)$ is the output, $f_i^1(\cdot)$ is a unity function of the i -th node, respectively, and N is the sampling iteration number.

Layer 2—Fuzzification Layer: The relationship in this layer is expressed as:

$$x_j^2(N) = y_i^1(N), i = 1 \tag{16}$$

$$net_j^2(N) = \left[\frac{(x_j^2(N) - c_j^2)^2}{(b_j^2)^2} \right] \tag{17}$$

$$y_j^2(N) = f_j^2(net_j^2(N)) = e^{-net_j^2(N)}, j = 1, 2, 3 \tag{18}$$

where $x_j^2(N) = y_j^1(N)$ is the input of the fuzzification layer, c_j^2 is the mean value, b_j^2 is the standard deviation, $net_j^2(N)$ is the Gaussian function, $f_j^2(\cdot)$ is the negative exponential function, and $y_j^2(N)$ is the output of the j -th node, respectively.

Layer 3—LSTM Layer: The relationship of this layer is expressed as follows:

$$x_k^3(N) = y_k^2(N) \tag{19}$$

$$f_k^3(N) = \sigma(w_{fk}^3 \cdot h_k^3(N-1) + u_{fk}^3 \cdot x_k^3(N) + b_{fk}^3) \tag{20}$$

$$u_k^3(N) = \sigma(w_{uk}^3 \cdot h_k^3(N-1) + u_{uk}^3 \cdot x_k^3(N) + b_{uk}^3) \tag{21}$$

$$a_k^3(N) = \tanh(w_{ak}^3 \cdot h_k^3(N-1) + u_{ak}^3 \cdot x_k^3(N) + b_{ak}^3) \tag{22}$$

$$c_k^3(N) = c_k^3(N-1) \circ f_k^3(N) + u_k^3(N) \circ a_k^3(N) \tag{23}$$

$$g_k^3(N) = \tanh(c_k^3(N)) \tag{24}$$

$$o_k^3(N) = \sigma(w_{ok}^3 \cdot h_k^3(N-1) + u_{ok}^3 \cdot x_k^3(N) + b_{ok}^3) \tag{25}$$

$$h_k^3(N) = o_k^3(N) \circ g_k^3(N) \tag{26}$$

$$net_k^3(N) = h_k^3(N) \tag{27}$$

$$y_k^3(N) = f_k^3(net_k^3(N)) = h_k^3(N), k = 1, 2, 3 \tag{28}$$

where

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{29}$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{30}$$

where $[w_{fk}^3, u_{fk}^3, b_{fk}^3]$, $[w_{uk}^3, u_{uk}^3, b_{uk}^3]$, $[w_{ak}^3, u_{ak}^3, b_{ak}^3]$, $[w_{ok}^3, u_{ok}^3, b_{ok}^3]$ are the weight and bias terms of the different LSTM parts, symbol \circ denotes dot product, $\tanh(z)$ and $\sigma(z)$ show nonlinear activation functions, $f_k^3(N)$ is the output of the forgetting gate, $u_k^3(N) \circ a_k^3(N)$ is the output of the input gate, $c_k^3(N)$ is the state value, $y_k^3(N)$ is the output at the N -th iteration of the k -th node, and $h_k^3(N)$ is the output of the output gate.

Layer 4—Defuzzification Layer: The relationship of this layer is expressed as:

$$x_k^4(N) = y_k^3(N) \tag{31}$$

$$net_j^4(N) = \frac{\sum_k w_{kl}^4 \cdot x_k^4(N)}{\sum_k x_k^4(N)} \tag{32}$$

$$y_l^4(N) = f_l^4(N) = net_l^4(N), l = 1, 2, 3 \tag{33}$$

where w_{kl}^4 is the weight of the fourth layer's l -th node connected with the k -th input, $net_l^4(N)$ is the network output, and $y_l^4(N)$ is the output of the l -th node.

Layer 5—Output Layer: The relationship of this layer is expressed as:

$$x_l^5(N) = y_l^4(N) \tag{34}$$

$$net_o^5(N) = \sum_l w_l^5 \cdot x_l^5(N) \tag{35}$$

$$y_o^5(N) = f_o^5(net_o^5(N)) = net_o^5(N), o = 1 \tag{36}$$

where w_l^5 is the weight related to the fifth layer output node and the l -th input, $net_o^5(N)$ is the network output, and $y_o^5(N)$ is the final output.

The learning mechanism of the NFNN is summarized as follows:

- (1) The system error is transmitted as the input to the fuzzy layer by the input layer.

- (2) Then, each neuron of the fuzzy layer employs a Gaussian function to make the input fuzzified, and the output is then passed to the LSTM layer for information extraction and mining.
- (3) The input of the LSTM layer and the feedback output of the previous state join with the input information channel and pass through three gates of forgetting, input, and output in turn.
- (4) The output of the LSTM layer is then passed to the defuzzification layer, using the weighted average method.
- (5) Finally, a weighted summation is accomplished on the defuzzification result to obtain the final output.

4. Adaptive Sliding Mode Controller Design Using the NFNN and Stability Analysis

Figure 3 gives the block diagram of the designed ASMC-NFNN, which mainly includes three parts: reference signal, proposed controller, and dynamic system. In the controller module, the NFNN took the error information of the system as input, used the gradient descent method for optimization, and adaptively learnt in order to obtain the unknown uncertainty of the system; SMC as the main body of the controller, combined with the estimated value output by the NFNN and robust term, gave the final effective control law. In addition, the control object was a system that had unmodeled dynamics and was subject to internal parameter changes and external disturbances. The control goal was not to rely on an accurate system model to achieve the task of accurate control under different disturbances. The following introduces the design of the proposed NFNN and ASMC-NFNN controller.

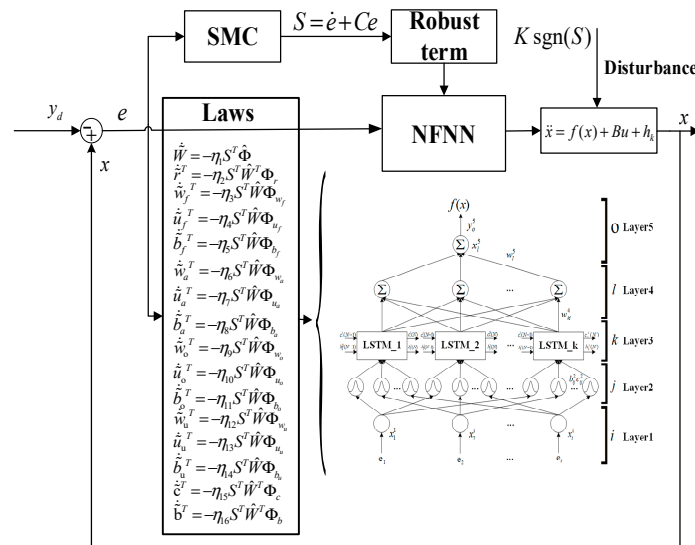


Figure 3. Block diagram of the proposed control system.

According to the universal approximation rule, there are optimal weights to use the output of the NFNN to estimate any smooth nonlinear function. Consequently, a NN is designed to identify the unknown terms of the system.

Assume that optimal weights $W^*, r^*, w_f^*, u_f^*, b_f^*, w_a^*, u_a^*, b_a^*, w_u^*, u_u^*, b_u^*, w_o^*, u_o^*, b_o^*, c^*, b^*$ exist that could estimate the unknown $f(X)$. This five-layer NFNN is given in Figure 1, written as:

$$f(X) = W^{*T} \Phi^* + \varepsilon \tag{37}$$

where $\Phi^* = \Phi^*(x, r^*, w_f^*, u_f^*, b_f^*, w_a^*, u_a^*, b_a^*, w_u^*, u_u^*, b_u^*, w_o^*, u_o^*, b_o^*, c^*, b^*)$, $W^* = [W_1^{5*} \ W_2^{5*} \ W_3^{5*}]^T$, Φ^* is the ideal output in the fourth layer with regards to the weights, W^* is the ideal weight generated from online learning, and ε is a reconstruction error, ensuring $|\varepsilon| \leq \varepsilon_b$, ε_b is a positive value.

Practically, the estimate of $f(X)$ by the NFNN is written as:

$$\hat{f}(X) = \hat{W}^T \hat{\Phi} \tag{38}$$

Then, the estimation error of $f(X)$ function is derived as:

$$\begin{aligned} f(X) - \hat{f}(X) &= W^{*T} \Phi^* - \hat{W}^T \hat{\Phi} + \varepsilon \\ &= W^{*T} (\hat{\Phi} + \tilde{\Phi}_2) - \hat{W}^T \hat{\Phi} + \varepsilon \\ &= W^{*T} \hat{\Phi} + W^{*T} \tilde{\Phi} - \hat{W}^T \hat{\Phi} + \varepsilon \\ &= \tilde{W}^T \hat{\Phi} + \hat{W}^T \tilde{\Phi} + \tilde{W}^T \tilde{\Phi} + \varepsilon \end{aligned} \tag{39}$$

where $\tilde{W}^T \tilde{\Phi} + \varepsilon = \varepsilon_0$ is the approximation error, $\tilde{W} = W^* - \hat{W}$, and $\tilde{\Phi} = \Phi^* - \hat{\Phi}$. Then, the Taylor expansion of $\tilde{\Phi}$ is applied to convert the nonlinear NFNN into a partially linear form as follows:

$$\begin{aligned} \tilde{\Phi} &= \frac{\partial \Phi}{\partial w_f} \Big|_{w_f=\hat{w}_f} (w_f^* - \hat{w}_f) + \frac{\partial \Phi}{\partial u_f} \Big|_{u_f=\hat{u}_f} (u_f^* - \hat{u}_f) + \frac{\partial \Phi}{\partial b_f} \Big|_{b_f=\hat{b}_f} (b_f^* - \hat{b}_f) \\ &\quad + \frac{\partial \Phi}{\partial w_a} \Big|_{w_a=\hat{w}_a} (w_a^* - \hat{w}_a) + \frac{\partial \Phi}{\partial u_a} \Big|_{u_a=\hat{u}_a} (u_a^* - \hat{u}_a) + \frac{\partial \Phi}{\partial b_a} \Big|_{b_a=\hat{b}_a} (b_a^* - \hat{b}_a) \\ &\quad + \frac{\partial \Phi}{\partial w_u} \Big|_{w_u=\hat{w}_u} (w_u^* - \hat{w}_u) + \frac{\partial \Phi}{\partial u_u} \Big|_{u_u=\hat{u}_u} (u_u^* - \hat{u}_u) + \frac{\partial \Phi}{\partial b_u} \Big|_{b_u=\hat{b}_u} (b_u^* - \hat{b}_u) \\ &\quad + \frac{\partial \Phi}{\partial w_o} \Big|_{w_o=\hat{w}_o} (w_o^* - \hat{w}_o) + \frac{\partial \Phi}{\partial u_o} \Big|_{u_o=\hat{u}_o} (u_o^* - \hat{u}_o) + \frac{\partial \Phi}{\partial b_o} \Big|_{b_o=\hat{b}_o} (b_o^* - \hat{b}_o) \\ &\quad + \frac{\partial \Phi}{\partial r} \Big|_{r=\hat{r}} (r^* - \hat{r}) + \frac{\partial \Phi}{\partial c} \Big|_{c=\hat{c}} (c^* - \hat{c}) + \frac{\partial \Phi}{\partial b} \Big|_{b=\hat{b}} (b^* - \hat{b}) + O_h \\ &= \Phi_{w_f} \cdot \tilde{w}_f + \Phi_{u_f} \cdot \tilde{u}_f + \Phi_{b_f} \cdot \tilde{b}_f + \Phi_{w_a} \cdot \tilde{w}_a + \Phi_{u_a} \cdot \tilde{u}_a + \Phi_{b_a} \cdot \tilde{b}_a \\ &\quad + \Phi_{w_u} \cdot \tilde{w}_u + \Phi_{u_u} \cdot \tilde{u}_u + \Phi_{b_u} \cdot \tilde{b}_u + \Phi_{w_o} \cdot \tilde{w}_o + \Phi_{u_o} \cdot \tilde{u}_o + \Phi_{b_o} \cdot \tilde{b}_o \\ &\quad + \Phi_r \cdot \tilde{r} + \Phi_c \cdot \tilde{c} + \Phi_b \cdot \tilde{b} + O_h \end{aligned} \tag{40}$$

where O_h is the higher-order term of the expansion and the partial terms, which can be calculated by the chain rule [36], are represented as:

$$\begin{aligned} \Phi_r &= \left[\begin{array}{ccc} \frac{\partial \Phi_1}{\partial w_{11}^4} & \dots & \frac{\partial \Phi_1}{\partial w_{33}^4} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Phi_3}{\partial w_{11}^4} & \dots & \frac{\partial \Phi_3}{\partial w_{33}^4} \end{array} \right] \Big|_{r=\hat{r}} \in R^{3 \times 9} & \Phi_b &= \left[\begin{array}{ccc} \frac{\partial \Phi_1}{\partial b_1^2} & \dots & \frac{\partial \Phi_1}{\partial b_3^2} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Phi_3}{\partial b_1^2} & \dots & \frac{\partial \Phi_3}{\partial b_3^2} \end{array} \right] \Big|_{b=\hat{b}} \in R^{3 \times 3} \\ \Phi_c &= \left[\begin{array}{ccc} \frac{\partial \Phi_1}{\partial c_1^2} & \dots & \frac{\partial \Phi_1}{\partial c_3^2} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Phi_3}{\partial c_1^2} & \dots & \frac{\partial \Phi_3}{\partial c_3^2} \end{array} \right] \Big|_{c=\hat{c}} \in R^{3 \times 3} & \Phi_{w_f} &= \left[\begin{array}{ccc} \frac{\partial \Phi_1}{\partial w_{f_1}^3} & \dots & \frac{\partial \Phi_1}{\partial w_{f_3}^3} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Phi_3}{\partial w_{f_1}^3} & \dots & \frac{\partial \Phi_3}{\partial w_{f_3}^3} \end{array} \right] \Big|_{w_f=\hat{w}_f} \in R^{3 \times 3} \end{aligned}$$

Then, from Equation (10), a practical controller is designed as follows:

$$u = \frac{1}{b(X)} (\hat{f}(X) + x_m^n) - \frac{1}{c_n b(X)} (c_1 \dot{e} + c_2 e^{(2)} + \dots + c_{n-1} e^{(n-1)}) - K \text{sgn}(S) \tag{41}$$

Theorem 1. Considering the nonlinear system (4), the proposed controller with the ASMC-NFNN strategy can be guaranteed to be asymptotically stable if the following conditions are satisfied:

- (1) The ASMC-NFNN controller is designed as in (41).
- (2) The updating laws of the NFNN are derived as in (42)–(57).

$$\dot{\tilde{W}} = \eta_1 S^T \hat{\Phi} \tag{42}$$

$$\dot{\tilde{r}}^T = \eta_2 S^T \hat{W}^T \Phi_r \quad (43)$$

$$\dot{\tilde{w}}_f^T = \eta_3 S^T \hat{W} \Phi_{w_f} \quad (44)$$

$$\dot{\tilde{u}}_f^T = \eta_4 S^T \hat{W} \Phi_{u_f} \quad (45)$$

$$\dot{\tilde{b}}_f^T = \eta_5 S^T \hat{W} \Phi_{b_f} \quad (46)$$

$$\dot{\tilde{w}}_a^T = \eta_6 S^T \hat{W} \Phi_{w_a} \quad (47)$$

$$\dot{\tilde{u}}_a^T = \eta_7 S^T \hat{W} \Phi_{u_a} \quad (48)$$

$$\dot{\tilde{b}}_a^T = \eta_8 S^T \hat{W} \Phi_{b_a} \quad (49)$$

$$\dot{\tilde{w}}_o^T = \eta_9 S^T \hat{W} \Phi_{w_o} \quad (50)$$

$$\dot{\tilde{u}}_o^T = \eta_{10} S^T \hat{W} \Phi_{u_o} \quad (51)$$

$$\dot{\tilde{b}}_o^T = \eta_{11} S^T \hat{W} \Phi_{b_o} \quad (52)$$

$$\dot{\tilde{w}}_u^T = \eta_{12} S^T \hat{W} \Phi_{w_u} \quad (53)$$

$$\dot{\tilde{u}}_u^T = \eta_{13} S^T \hat{W} \Phi_{u_u} \quad (54)$$

$$\dot{\tilde{b}}_u^T = \eta_{14} S^T \hat{W} \Phi_{b_u} \quad (55)$$

$$\dot{\tilde{c}}^T = \eta_{15} S^T \hat{W}^T \Phi_c \quad (56)$$

$$\dot{\tilde{b}}^T = \eta_{16} S^T \hat{W}^T \Phi_b \quad (57)$$

where $\eta_1, \eta_2, \dots, \eta_{16}$ are learning rate parameters, both of which are positive constants.

Remark 3. Compared with the existing research, the proposed control strategy had a better control performance. First, compared to traditional sliding mode control, the ASMC-NFNN had the advantage of less chattering because, in the case of unknown system disturbances, traditional sliding mode control often relies on high-gain switching gain to achieve disturbance compensation. Although this could bring a good steady-state performance, it could cause high-frequency output chattering, which could bring huge adverse effects to the system. On the contrary, the proposed method firstly relied on the learning and estimation ability of the neural network to realize the active compensation of the disturbance, thereby reducing the burden of the sliding mode on the disturbance, and reducing the system chattering while improving the performance. Second, the proposed strategy had better dynamic performance and robustness than other neural network-based sliding mode control strategies because it adopted a novel neural network with an LSTM structure. As mentioned above, in theory, it had a selective forgetting mechanism, avoided the problem of gradient disappearance, and had the advantages of a strong learning ability. Therefore, it could compensate for the unknown time-varying uncertainty of the system faster and more accurately, and, thus, had better dynamic performance. The subsequent simulation comparison results also showed that the proposed control strategy could cope with larger parameter changes and had better dynamic performance.

Proof. The following Lyapunov function is designed as:

$$\begin{aligned}
 V_2 = & \frac{1}{2}S^T S + \frac{1}{2\eta_1} \text{tr}(\tilde{W}^T \tilde{W}) + \frac{1}{2\eta_2} \text{tr}(\tilde{r}^T \tilde{r}) + \frac{1}{2\eta_3} \text{tr}(\tilde{w}_f^T \tilde{w}_f) \\
 & + \frac{1}{2\eta_4} \text{tr}(\tilde{u}_f^T \tilde{u}_f) + \frac{1}{2\eta_5} \text{tr}(\tilde{b}_f^T \tilde{b}_f) + \frac{1}{2\eta_6} \text{tr}(\tilde{w}_a^T \tilde{w}_a) \\
 & + \frac{1}{2\eta_7} \text{tr}(\tilde{u}_a^T \tilde{u}_a) + \frac{1}{2\eta_8} \text{tr}(\tilde{b}_a^T \tilde{b}_a) + \frac{1}{2\eta_9} \text{tr}(\tilde{w}_o^T \tilde{w}_o) \\
 & + \frac{1}{2\eta_{10}} \text{tr}(\tilde{u}_o^T \tilde{u}_o) + \frac{1}{2\eta_{11}} \text{tr}(\tilde{b}_o^T \tilde{b}_o) + \frac{1}{2\eta_{12}} \text{tr}(\tilde{w}_u^T \tilde{w}_u) \\
 & + \frac{1}{2\eta_{13}} \text{tr}(\tilde{u}_u^T \tilde{u}_u) + \frac{1}{2\eta_{14}} \text{tr}(\tilde{b}_u^T \tilde{b}_u) + \frac{1}{2\eta_{15}} \text{tr}(\tilde{c}^T \tilde{c}) + \frac{1}{2\eta_{16}} \text{tr}(\tilde{b}^T \tilde{b})
 \end{aligned} \tag{58}$$

where V_2 is positive definite. The last 16 items in Equation (58) are denoted as M . \square

Remark 4. The designed Lyapunov function not only included the quadratic form of the systematic error, but also the quadratic form of the estimation error of the neural network weight, which not only ensured the convergence of the systematic error, but also ensured that the parameter error was small. In addition, because the learning rate of the weight was relatively large, the quadratic term of the system error had a larger proportional coefficient than the weight error, which meant that the convergence of the system error was guaranteed to a greater extent. In addition, compared with the traditional neural network method using gradient descent to optimize the error, this paper incorporated the quadratic form of the weight error into the design of the Lyapunov function, so that the adaptive rate of the neural network could be reversely derived through the stability proof. Although the form of the two was found to be similar after derivation, it was clear that the method proposed in this paper had more theoretical support, which was a unique contribution of this paper.

Taking the derivative of (58) and then substituting (8) into it the following is obtained:

$$\begin{aligned}
 \dot{V}_2 &= S^T \dot{S} + \dot{M} \\
 &= S^T [c_1 \dot{e} + c_2 e^{(2)} + \dots + c_{n-1} e^{(n-1)} + c_n (b(X)u - f(X) + \delta - x_m^n)] + \dot{M} \\
 &= S^T [\hat{f}(X) - f(X) + \delta - c_n b(X) \text{Ksgn}(S)] + \dot{M} \\
 &= S^T [-W^{*T} \Phi_2^* - \varepsilon + \hat{W}^T \hat{\Phi}_2 + \delta - c_n b(X) \text{Ksgn}(S)] + \dot{M} \\
 &= S^T [-\tilde{W}^T \hat{\Phi}_2 - \hat{W}^T \tilde{\Phi}_2 - \varepsilon_0 + \delta - c_n b(X) \text{Ksgn}(S)] + \dot{M}
 \end{aligned} \tag{59}$$

Substituting Taylor's expansion (40) into (59) yields

$$\begin{aligned}
 \dot{V}_2 &= -S^T \tilde{W}^T \hat{\Phi} - S^T \hat{W}^T (\Phi_{w_f} \cdot \tilde{w}_f + \Phi_{u_f} \cdot \tilde{u}_f + \Phi_{b_f} \cdot \tilde{b}_f + \Phi_{w_a} \cdot \tilde{w}_a \\
 &+ \Phi_{u_a} \cdot \tilde{u}_a + \Phi_{b_a} \cdot \tilde{b}_a + \Phi_{w_u} \cdot \tilde{w}_u + \Phi_{u_u} \cdot \tilde{u}_u + \Phi_{b_u} \cdot \tilde{b}_u + \Phi_{w_o} \cdot \tilde{w}_o \\
 &+ \Phi_{u_o} \cdot \tilde{u}_o + \Phi_{b_o} \cdot \tilde{b}_o + \Phi_r \cdot \tilde{r} + \Phi_c \cdot \tilde{c} + \Phi_b \cdot \tilde{b} + O_h) \\
 &+ S^T [-\varepsilon_0 + \delta - c_n b(X) \text{Ksgn}(S)] + \frac{1}{\eta_1} \text{tr}(\tilde{W}^T \dot{\tilde{W}}) + \frac{1}{\eta_2} \text{tr}(\dot{\tilde{r}}^T \tilde{r}) \\
 &+ \frac{1}{\eta_3} \text{tr}(\dot{\tilde{w}}_f^T \tilde{w}_f) + \frac{1}{\eta_4} \text{tr}(\dot{\tilde{u}}_f^T \tilde{u}_f) + \frac{1}{\eta_5} \text{tr}(\dot{\tilde{b}}_f^T \tilde{b}_f) + \frac{1}{\eta_6} \text{tr}(\dot{\tilde{w}}_a^T \tilde{w}_a) \\
 &+ \frac{1}{\eta_7} \text{tr}(\dot{\tilde{u}}_a^T \tilde{u}_a) + \frac{1}{\eta_8} \text{tr}(\dot{\tilde{b}}_a^T \tilde{b}_a) + \frac{1}{\eta_9} \text{tr}(\dot{\tilde{w}}_o^T \tilde{w}_o) + \frac{1}{\eta_{10}} \text{tr}(\dot{\tilde{u}}_o^T \tilde{u}_o) \\
 &+ \frac{1}{\eta_{11}} \text{tr}(\dot{\tilde{b}}_o^T \tilde{b}_o) + \frac{1}{\eta_{12}} \text{tr}(\dot{\tilde{w}}_u^T \tilde{w}_u) + \frac{1}{\eta_{13}} \text{tr}(\dot{\tilde{u}}_u^T \tilde{u}_u) + \frac{1}{\eta_{14}} \text{tr}(\dot{\tilde{b}}_u^T \tilde{b}_u) \\
 &+ \frac{1}{\eta_{15}} \text{tr}(\dot{\tilde{c}}^T \tilde{c}) + \frac{1}{\eta_{16}} \text{tr}(\dot{\tilde{b}}^T \tilde{b})
 \end{aligned} \tag{60}$$

Finally, substituting the updating laws (42)~(57) into (60) the following is obtained:

$$\begin{aligned} \dot{V}_2 &= S^T[\delta - \varepsilon_0 + O_{ho} - c_n b(X)K \text{sgn}(S)] \\ &= S^T(\delta - \varepsilon_0 + O_{ho}) - c_n b(X)K|S| \\ &\leq |S|(\delta_b - \varepsilon_0 + O_{ho}) - c_n b_l K|S| \\ &\leq |S|(\delta_b + \varepsilon_E + O_E - c_n b_l K) \end{aligned} \tag{61}$$

where $O_{ho} = \hat{w}^T O_h$. Suppose ε_0 and O_{ho} have upper bounds ε_E and O_E respectively as $|\varepsilon_0| \leq \varepsilon_E, |O_{ho}| \leq O_E$. So, if $K \geq (\delta_b + \varepsilon_E + O_E) / c_n b_l$, $\dot{V}_2 \leq 0$.

Because \dot{V}_2 is a semi-negative definite, V_2, S are bounded. From inequality $\dot{V}_2 \leq |S|(\delta_b + \varepsilon_E + O_E - c_n b_l K)$, S is integrated as $\int_0^t |S| dt \leq \frac{1}{\delta_b + \varepsilon_E + O_E - c_n b_l K} [V_2(t) - V_2(0)]$. $V_2(0)$ is bounded and $0 \leq V_2(t) \leq V_2(0)$, $\lim_{t \rightarrow \infty} \int_0^t |S| dt$ is bounded. From Barbalat's lemma, S and e asymptotically converge to zero as $t \rightarrow \infty$.

5. Simulation Study

To verify the effectiveness of the proposed ASMC-NFNN algorithm, a simulation experiment was performed using Matlab/Simulink with a single-phase active power filter (APF), as in Figure 4. In the simulation experiment, the computer system was 64-bit, the CPU was i7-6500 U (2.5 GHz), and the Matlab software version was 2019b. The shunt single-phase APF circuit model had three main components: grid voltage, nonlinear load, and APF main circuit.

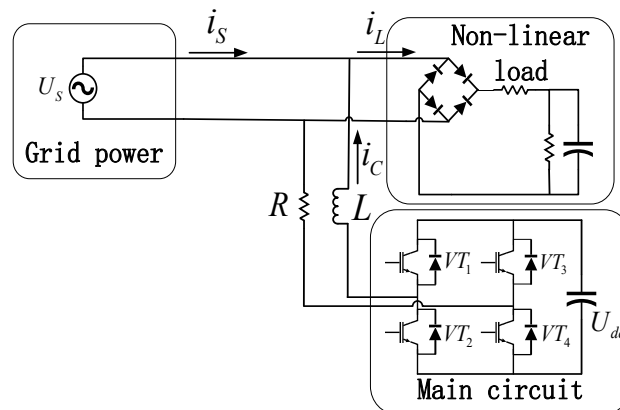


Figure 4. Circuit model of a single-phase APF.

In fact, the APF controller included three parts: harmonic detection module, DC side voltage control module, and compensation current tracking control module. The control goal was to force the compensation current to follow the reference current quickly and accurately. On the one hand, the harmonic detection module was implemented by a single-phase fast harmonic detection algorithm, which could obtain the reference current in real time. On the other hand, the control of DC side voltage was realized by traditional PID control due to low control requirements and low control difficulty, and voltage stability was achieved by superimposing the output of the PID controller into the reference information. Therefore, the voltage on the DC side was stable and could be regarded as a constant.

From circuit theory, the following equations are obtained:

$$\begin{cases} U_s = L \frac{di_c}{dt} + Ri_c + uU_{dc} \\ ui_c = C \frac{dU_{dc}}{dt} \end{cases} \tag{62}$$

where U_s is a grid voltage, i_c is a compensation current, U_{dc} is a capacitor voltage in DC side, and L and R are inductance and resistance. respectively. u is defined as

$$u = \begin{cases} 1 & VT_1, VT_4 \text{ on}, VT_2, VT_3 \text{ off} \\ -1 & VT_2, VT_3 \text{ on}, VT_1, VT_4 \text{ off} \end{cases} \tag{63}$$

In this paper, the state equation for the compensation current was studied since a PID controller applied in the DC side voltage made it easy to obtain the voltage requirements. The compensation current is derived as

$$\dot{i}_c = -\frac{R}{L}i_c + \frac{U_s}{L} - \frac{U_{dc}}{L}u \tag{64}$$

However, under system uncertainties and disturbances, the state equation for the compensation current is further revised as

$$\dot{i}_c = -\frac{R + \Delta R}{L + \Delta L}i_c + \frac{U_s}{L + \Delta L} - \frac{U_{dc}}{L + \Delta L}u + g \tag{65}$$

where ΔR and ΔL are the uncertainties of R and L respectively, and g is the other uncertain component. Further, Equation (65) is rewritten as:

$$\dot{i}_c = -\frac{R}{L}i_c + \frac{U_s}{L} - \frac{U_{dc}}{L}u + h_0 \tag{66}$$

where h_0 is the lumped uncertainty bounded by H_0 , as $|h_0| \leq H_0$.

Taking the derivative of Equation (66) one gets

$$\begin{aligned} \ddot{i}_c &= -\frac{R}{L}\dot{i}_c + \frac{\dot{U}_s}{L} - \frac{\dot{U}_{dc}}{L}u + \dot{h}_0 \\ &= -\frac{R}{L}\left(-\frac{R}{L}i_c + \frac{U_s}{L} - \frac{U_{dc}}{L}u + h\right) + \frac{\dot{U}_s}{L} - \frac{\dot{U}_{dc}}{L}u + \dot{h}_0 \\ &= \frac{R^2}{L^2}i_c + \frac{\dot{U}_s}{L} - \frac{R}{L^2}U_s + \frac{R}{L^2}U_{dc}u - \frac{R}{L}h - \frac{\dot{U}_{dc}}{L}u + \dot{h}_0 \\ &= \frac{R^2}{L^2}i_c + \frac{\dot{U}_s}{L} - \frac{R}{L^2}U_s + \frac{R}{L^2}U_{dc}u - \frac{R}{L}h + \dot{h}_0 \end{aligned} \tag{67}$$

Then, the second-order dynamic equation is obtained as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -f(x) + Bu + h_k \end{cases} \tag{68}$$

where x_1 represents i_c , B represents $R/L^2 * U_{dc}$, which is a known constant, $f(x)$ represents $-R^2/L^2 * i_c - \dot{U}_s/L + R/L^2 * U_s$, which is an unknown function whose exact value is difficult to obtain, h_k represents $-R/L * h + \dot{h}_0$, and h_k is the lumped uncertainty, with an upper bound H_k , as $|h_k| \leq H_k$.

The parameters in the system simulation are explained in Table 1. The parameters of the PID controller and the ASMC-NFNN controller are given as

$$K_p = 0.15, C = 17,200, K = 2592$$

In the NFNN, learning rates were $\eta_1, \eta_2 = 2.5 \times 10^{-3}, \eta_3, \eta_4, \dots, \eta_{16} = 5$, the initial values of the center and width in Gaussian function were selected as $c^2 = [-0.075 \ 0 \ 0.075]$, $b^2 = [1 \ 1 \ 1]$, and the initial weights were chosen to be 1, except for the initial weights $r^4 = [0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5 \ 0.6 \ 0.7 \ 0.8 \ 0.9]$ in the fourth layer and $w^5 = [0.9 \ 1.0 \ 1.1]$ in the fifth layer.

Table 1. Parameters in simulation.

Parameter	Value
Grid voltage and frequency	24 V/50 Hz
Nonlinear load in steady state	$R1 = 5 \Omega, R2 = 15 \Omega, C = 1 \times 10^{-3} \text{ F}$
Additional nonlinear loads in parallel	$R1 = 15 \Omega, R2 = 15 \Omega, C = 1 \times 10^{-3} \text{ F}$
Main circuit parameter	$L = 18 \times 10^{-3} \text{ H}, R = 1 \Omega, V_{\text{ref}} = 50 \text{ V}$
Sampling time	$T_s = 1 \times 10^{-5} \text{ s}$

Simulation verification included three aspects: (1) steady state response simulation for harmonic compensation; (2) dynamic response simulation and comparison with recurrent FNN; (3) parameter variations simulation and comparison.

5.1. Steady–State Response

In the simulation, the harmonic compensation was set to be added in the APF system from 0.05 s. Figure 5 gives the steady-state response of the system under the proposed ASMC–NFNN. The waveform diagrams were load current, power supply current, compensation current tracking curve, and tracking error figure, in order. As can be seen from the load current curve in Figure 5, the load had serious nonlinear distortion and caused serious harmonic distortion in the power supply current.

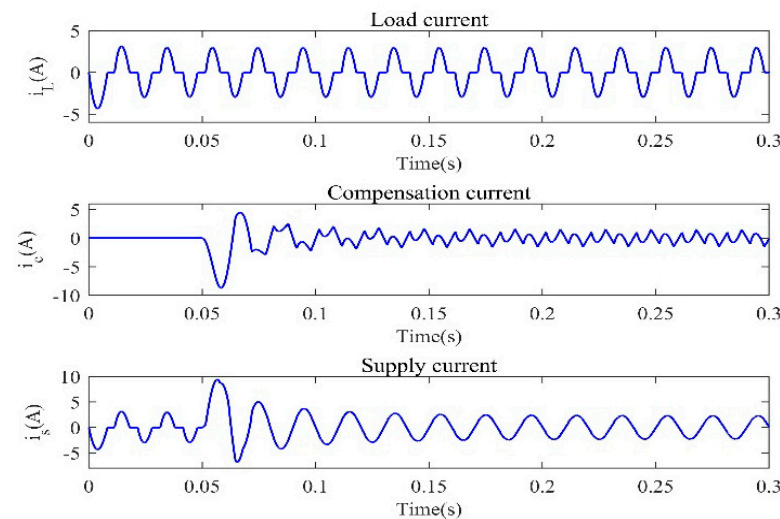


Figure 5. Current waveforms of steady–state response using ASMC–NFNN from top to bottom: load, compensation, and source currents.

The degree of harmonic distortion could be seen from the spectrum analysis chart in Figure 6. In addition to the fundamental wave, there were many harmonics, and the total harmonic distortion (THD) rate reached 35.07%. However, after APF started to control at 0.05 s, the harmonics were suppressed in a short time. From Figure 7, the THD was reduced to 2.3%. Therefore, APF using the ASMC–NFNN controller could well purify harmonic pollution. As shown in Figure 8, the compensation current curve and the reference current curve almost completely coincided after a short time, and the tracking error was also nearly zero, which reflected the high accuracy of current tracking, fast response, and good harmonic compensation effect.

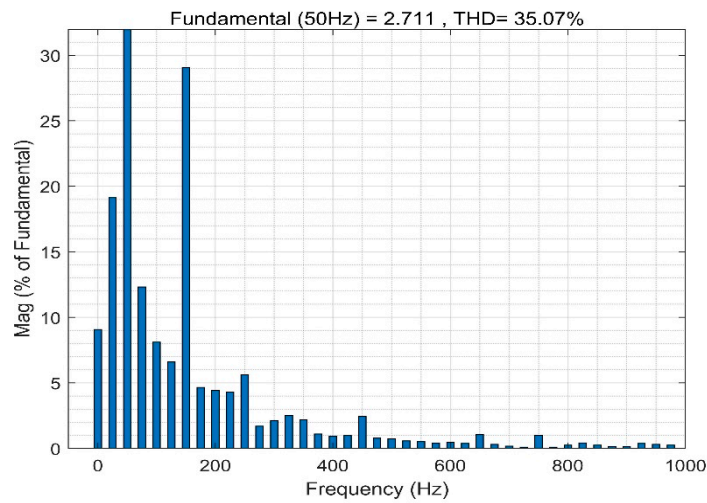


Figure 6. Harmonic spectrum of power supply current before compensation.

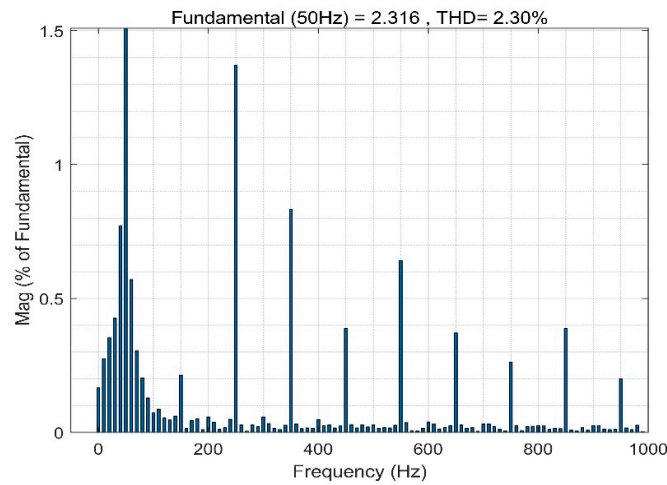


Figure 7. Harmonic spectrum of power supply current in steady state.

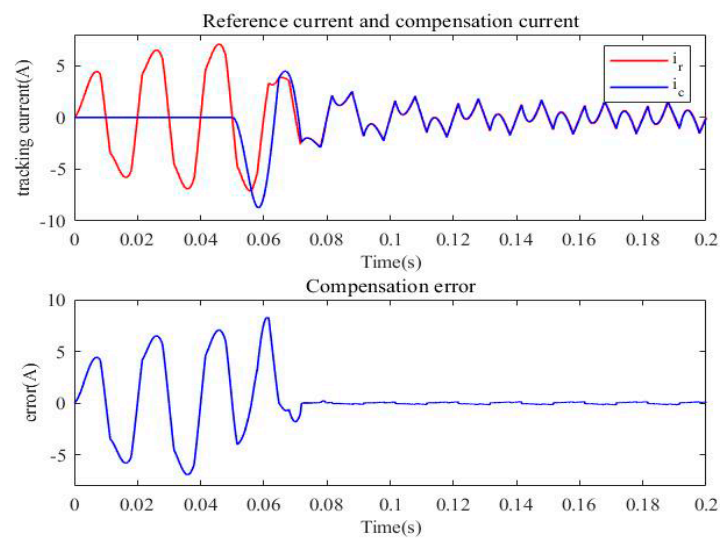


Figure 8. Current tracking and error waveforms of steady-state response using ASMC-NFNN from top to bottom: current tracking and tracking error curve.

Figures 9–12 give adaptive adjustment curves of some parameters of the NFNN, which did not need to be adjusted manually. After adaptive learning, convergence was achieved

in a short time and the system response could also achieve good results. This showed that the NFNN had an excellent adaptive adjustment performance and stability.

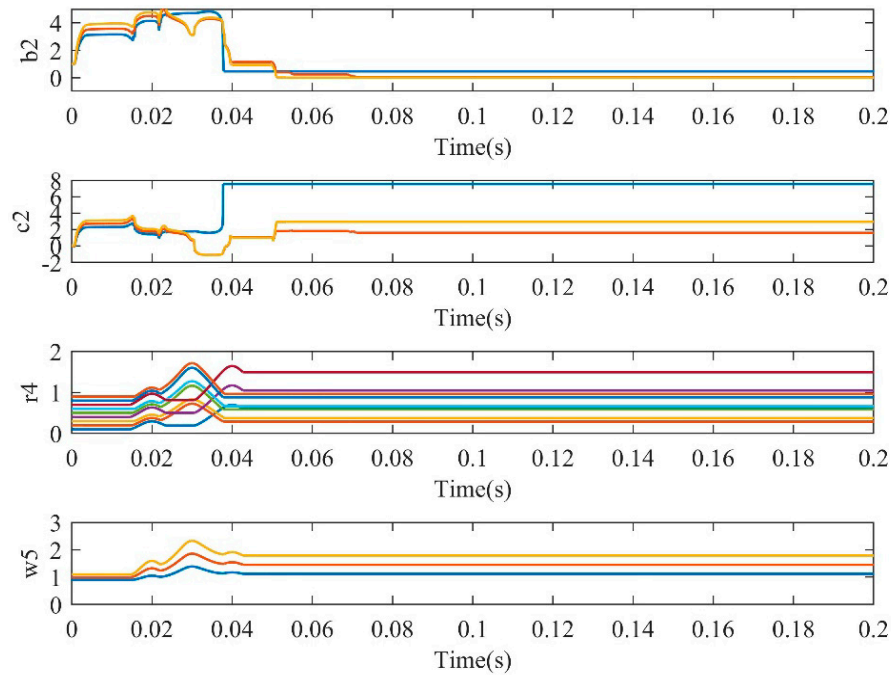


Figure 9. The first group of weights of adaptive learning curves.

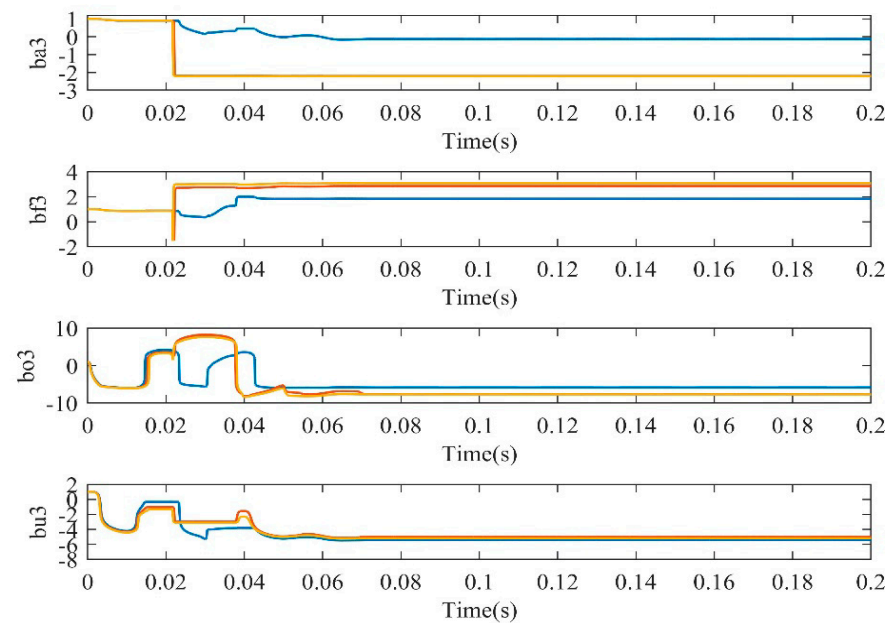


Figure 10. The second group of weights of adaptive learning curves.

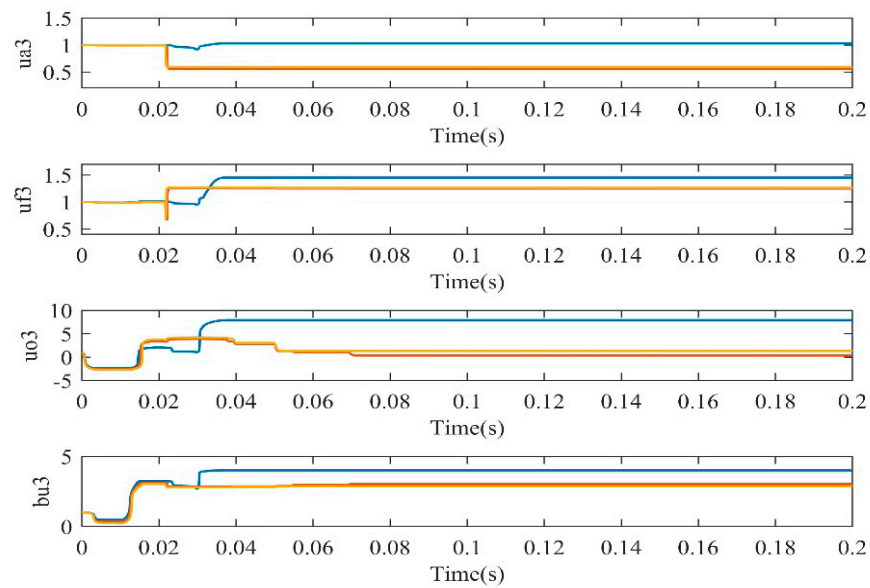


Figure 11. The third group of weights of adaptive learning curves.

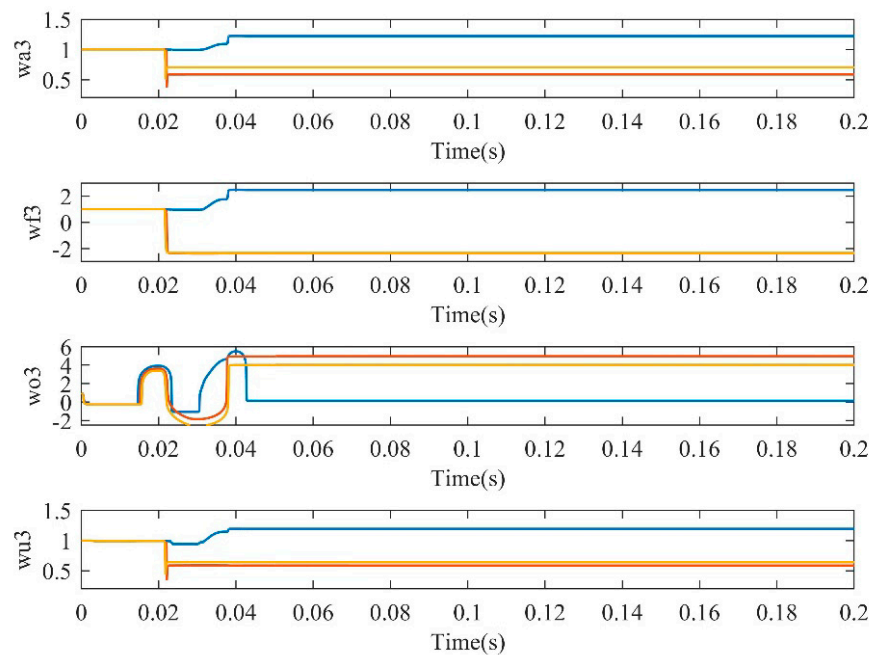


Figure 12. The fourth group of weights of adaptive learning curves.

5.2. Dynamic Response Simulation and Comparison

To verify the ability of APF to compensate for harmonics under sudden load changes using ASMC–NFNN, sudden load increase and decrease experiments were designed and the comparison simulation with ASMC–RFNN was given.

In the dynamic simulation, the APF was connected to the system at 0.05 s, and a nonlinear load was added at 0.3 s and subtracted at 0.6 s. In addition, the parameters of the increased load are given in Table 1. The dynamic response of the APF system was observed when the load changed. The dynamic response waveforms of the supply current using ASMC–NFNN is shown in Figure 13, which shows that no matter when the load was increased in 0.3 s or the load was decreased in 0.6 s, the power current returned to a sinusoidal steady state after a short adjustment, showing that the proposed controller worked well under load changes with good dynamic properties.

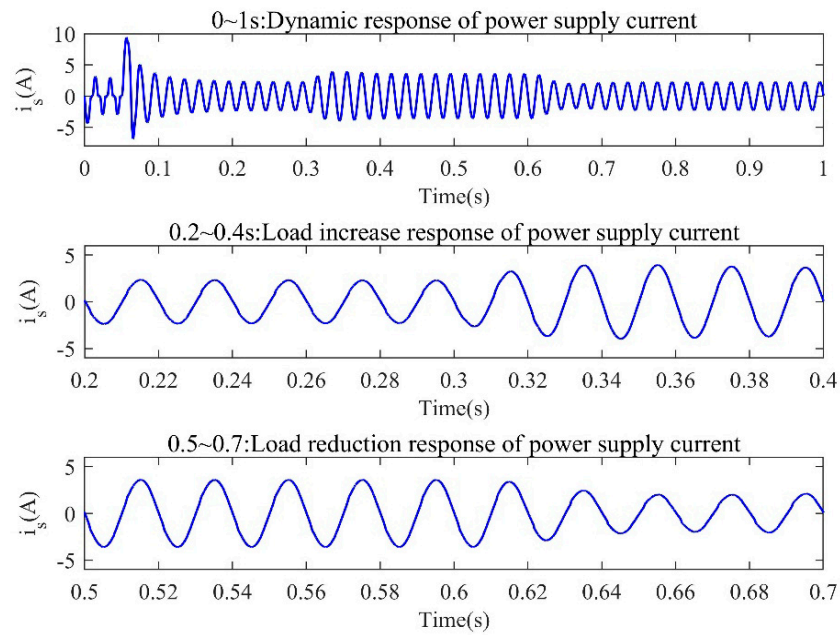


Figure 13. Dynamic response waveforms of power supply current using ASMC–NFNN (load increases at 0.3 s, load decreases at 0.6 s).

Moreover, the voltage control curve on the DC side is shown in Figure 14, verifying the previous assumption that the DC side voltage was stable.

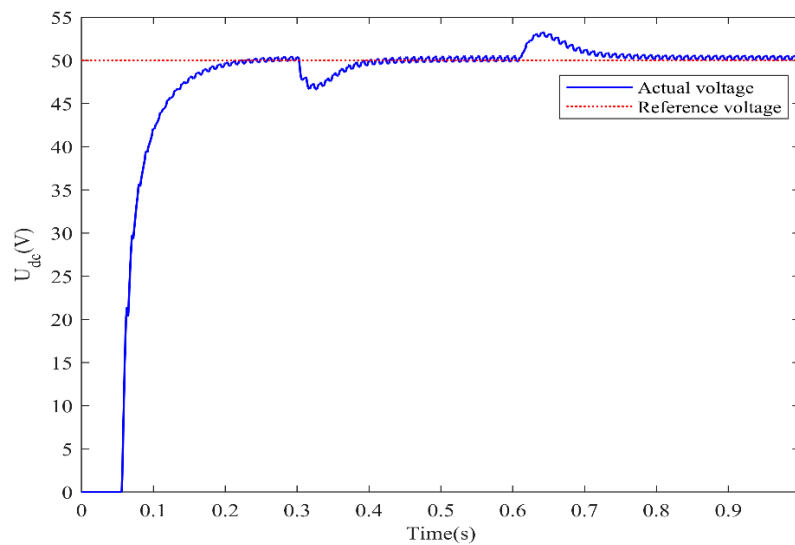


Figure 14. DC side voltage curve.

Moreover, Figure 15 shows the current tracking curves under the two methods. In both methods, the compensation current could track the reference current when the load changed. Moreover, it was roughly seen that the tracking curve of the comparison method was over-tracked and covered other curves, so its tracking compensation performance was worse.

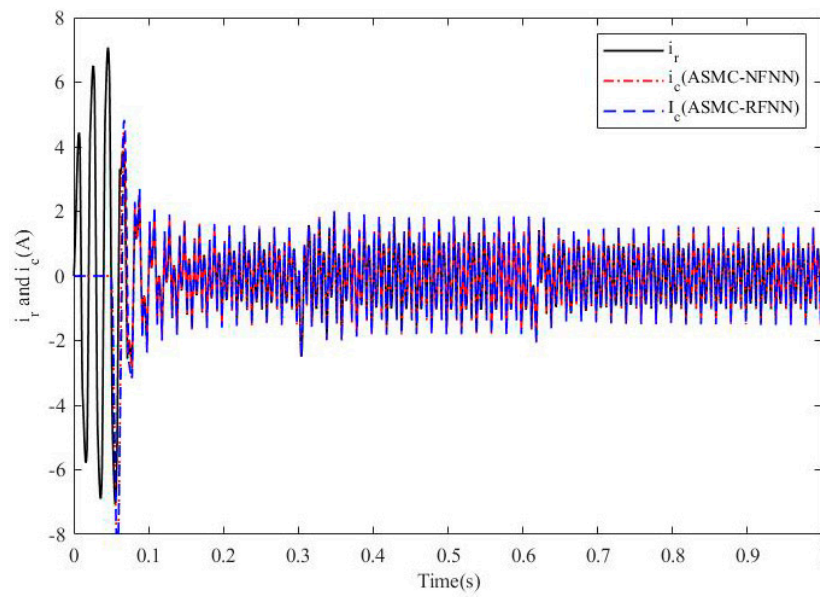


Figure 15. Current tracking curves (load increases at 0.3 s, load decreases at 0.6 s).

In order to see a more obvious comparison, the overall tracking error comparison curves are given in Figure 16, and the partial enlarged comparison curves at the load change point are given in Figure 17. It was clearly seen from Figure 17 that the error of the proposed method (red curve) at the two load change nodes was closer to 0 than the comparison method (blue curve). Therefore, the proposed controller with NFNN had better dynamic performance than the RFNN.

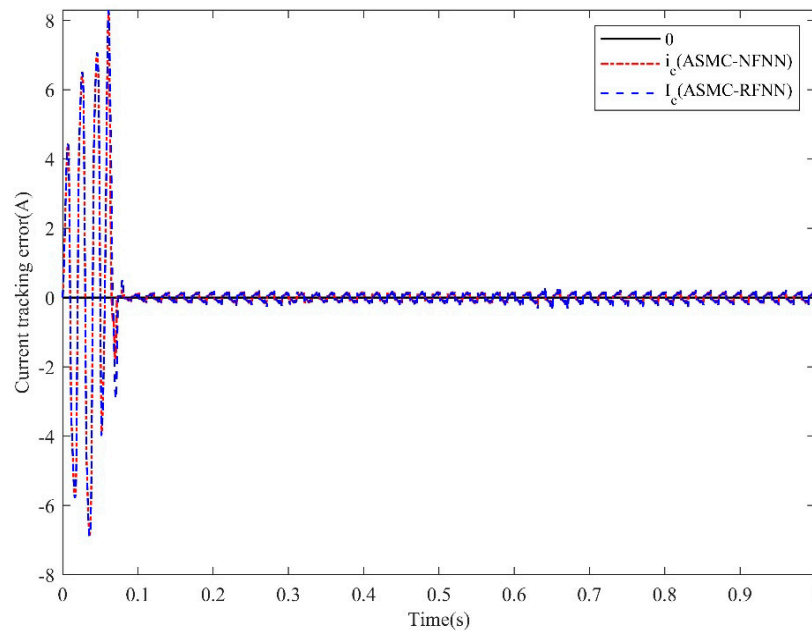


Figure 16. Current tracking error curves (load increases at 0.3 s, load decreases at 0.6 s).

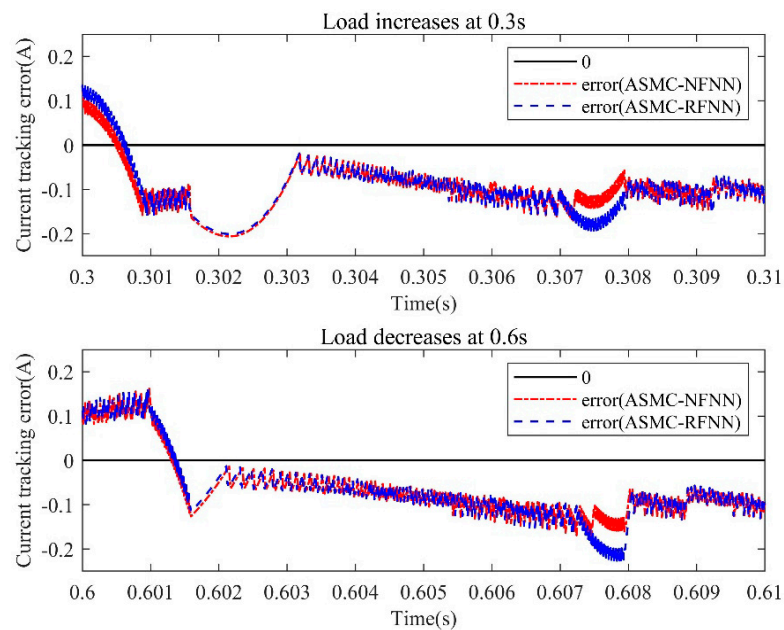


Figure 17. Enlarged curves of current tracking error (load increases at 0.3 s, load decreases at 0.6 s).

Several performance indicators were used to analyze the proposed NFNN-based controller and the comparative RFNN-based controller. The performance comparison results of THD are shown in Table 2. In three experimental states, the ASMC-NFNN had a smaller THD than the ASMC-RFNN. In addition, compared with the dynamic terminal sliding mode controller using a double hidden layer recurrent neural network (DTSMC-DHLRNN) proposed in Ref. [40], the THD performance of ASMC-NFNN in steady state was 0.66% better. In addition, the comparison results of some commonly used performance indicators are given in Table 3, showing the proposed controller was superior to the comparison method in various error performance indicators. However, in the indicator of simulation calculation time, because the proposed method had a more complicated structure, the calculation time was slightly longer than the comparison method. In fact, the update complexity per weight and time step of the LSTM algorithm was, essentially, that of BPTT, namely $O(1)$, and LSTM was local in both space and time [36]. Therefore, the extra complexity introduced by LSTM was not high and, fortunately, with the upgrade in computing power, the subtle difference in computing time was no longer a serious problem.

Table 2. THD comparison in simulation.

State/Strategy	THD of ASMC-NFNN	THD of ASMC-RFNN	THD of DTSMC-DHLRNN [40]
Steady state	2.30%	2.91%	2.96%
Load increase	1.58%	1.61%	2.64%
Load decrease	2.47%	3.24%	2.96%

Table 3. Other performance index comparisons in simulation.

Controller/Index	RMSE	ITSE	ITAE	Calculation Time (s)
ASMC-NFNN	0.0897	1.2166	14.7448	19
ASMC-RFNN	0.0901	2.1508	27.4268	12

5.3. Parameter Variations Simulation and Comparison

Practically, it was not easy to obtain accurate parameters of the controlled object, especially in the power system occasions, as the component parameters would also dynamically change. For example, the aging of the resistance made the resistance value larger,

and the inductance would change subject to environmental factors, such as temperature and magnetic field. The changes in the internal parameters of these systems were called internal disturbances, which would make the performance of general controllers worse and even difficult to use in practice. Therefore, this section studies the robustness of the proposed controller under parameter changes and compares it with the other two methods. The selected variable parameters were the resistance and inductance values on the APF main circuit.

Figure 18 shows the change curve using ASMC–NFNN of the steady–state THD value with different degrees of inductance attenuation. When the inductance attenuation percentage was small, the THD hardly changed. When the inductance attenuation degree gradually became larger, the value and the change range of THD also increased continuously. This showed that the inductance parameter had a great influence on the system performance. However, it could be seen from the figure that even if the inductance was attenuated by 40%, the THD of the power supply current was still below 5% under the proposed method, which could still meet the application requirements. Therefore, the proposed method had a large tolerance space for inductance parameters, and had good robustness in terms of inductance parameter changes.

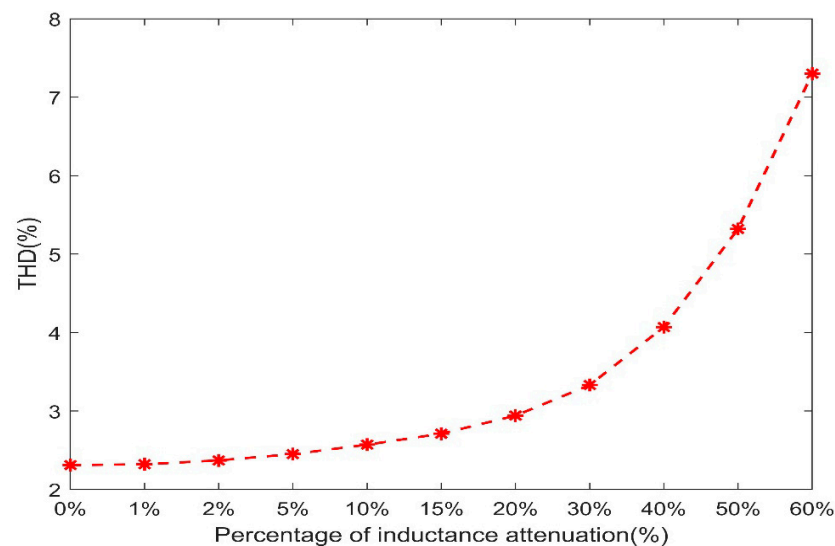


Figure 18. Performance curve of inductance change using ASMCN–FNN.

In addition, the steady–state THD comparison chart of the three methods under inductance attenuation is given in Figure 19, showing that the proposed strategy had the smallest THD variation and the best parameter tolerance. The method based on ASMC–RFNN had a small THD change when the inductance attenuation was small; when the inductance attenuation was greater than 5%, the THD increased sharply; after the inductance attenuation was 30%, the THD even exceeded the ordinary sliding mode controller. The above shows that when the inductance attenuation was greater than 30%, the RFNN network of the ASMC–RFNN method fell into disorder and failed to learn the system parameters correctly. At the same time, it proved that the NFNN had a better learning ability and adaptability than the RFNN. On the other hand, when the inductance parameters were not particularly large, the proposed algorithm with the NN had greater tolerance to inductance parameters and a smaller THD change trend than ordinary sliding mode controllers, showing the advantage of good robustness.

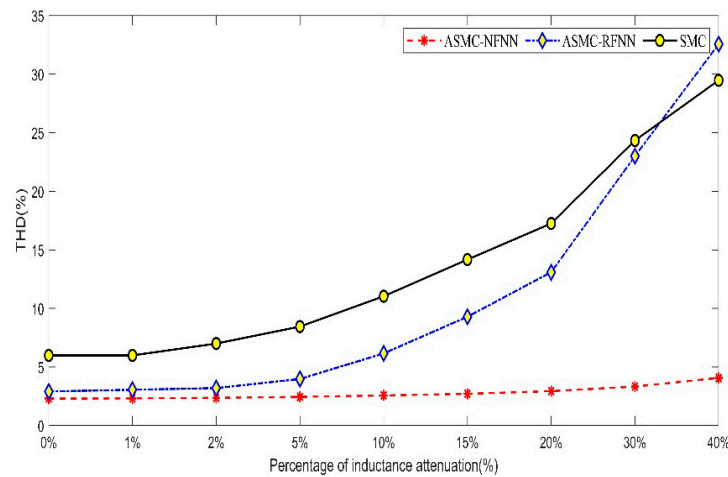


Figure 19. Performance comparison curve of inductance parameter changes.

At the same time, the simulation comparison chart of resistance variation is given in Figure 20, showing the change in the resistance had a very small effect on the system performance. The SMC-based method could tolerate an increase of 3.4 times in resistance, while the proposed ASMC-NFNN could tolerate an increase of 3.6 times in resistance. In general, changes in resistance parameters had a limited impact on the system, and ASMC-NFNN methods had better performances, to some extent.

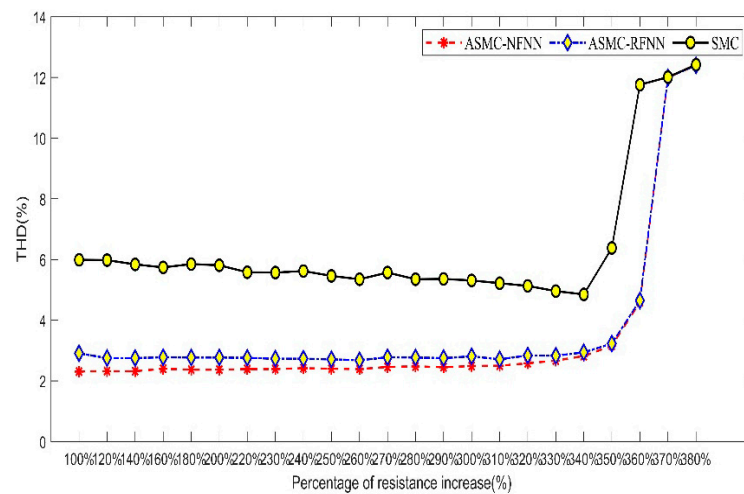


Figure 20. Performance comparison curve of resistance parameter changes.

6. Conclusions

In this paper, an ASMC-NFNN strategy was studied for a general class of dynamic systems with unknown uncertainties. Considering the existence of time-varying unknown uncertainty perturbations in such systems, an NFNN with an LSTM structure was developed to approximate the system uncertainty where the LSTM structure had a special gating unit that could selectively forget and remember, which was suitable for long-term dependent learning problems. In addition, a sliding mode controller was given to implement the tracking control of nonlinear systems, ensuring high tracking accuracy and fast response speed under estimation error and external disturbances. An NFNN online learning algorithm was derived and all the parameters of the NFNN were guaranteed to converge under the adaptive laws. The proposed control strategy were verified on the second-order single-phase APF system, and the numerical experimental results showed that it had better steady-state and dynamic properties than other control methods, and also had better robustness in the presence of parameter changes. Considering that the control strategy was universally designed, and the algorithm integrating neural network

and sliding mode were very suitable for power electronic control, the strategy could be extended to the control of a series of similar electronic power systems, and could achieve superior results. In future research, reducing the computational complexity of neural networks and the difficulty of parameter adjustment will be a hot research direction. In addition, the use of more advanced adaptive super-twisting sliding modes is also expected to be a promising research direction.

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

Funding: This work is support by National Science Foundation of China (Grant No. 61873085).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

- Schoukens, J.; Nemeth, J.G.; Vandersteen, G.; Pintelon, R.; Crama, P. Linearization of Nonlinear Dynamic Systems. *IEEE Trans. Instrum. Meas.* **2004**, *53*, 1245–1248. [[CrossRef](#)]
- Hotz, M.; Vogel, C. Linearization of Time-Varying Nonlinear Systems Using a Modified Linear Iterative Method. *IEEE Trans. Signal Process.* **2014**, *62*, 2566–2579. [[CrossRef](#)]
- Incremona, G.P.; Rubagotti, M.; Ferrara, A. Sliding Mode Control of Constrained Nonlinear Systems. *IEEE Trans. Autom. Control* **2017**, *62*, 2965–2972. [[CrossRef](#)]
- Bartolini, G.; Punta, E.; Zolezzi, T. Simplex Methods for Nonlinear Uncertain Sliding-Mode Control. *IEEE Trans. Autom. Control* **2004**, *49*, 922–933. [[CrossRef](#)]
- Zhang, C.; Yang, J.; Yan, Y.; Fridman, L.; Li, S. Semiglobal Finite-Time Trajectory Tracking Realization for Disturbed Nonlinear Systems via Higher-Order Sliding Modes. *IEEE Trans. Autom. Control* **2020**, *65*, 2185–2191. [[CrossRef](#)]
- Ding, S.; Ma, K.; Li, S. A New Second-Order Sliding Mode and Its Application to Nonlinear Constrained Systems. *IEEE Trans. Autom. Control* **2019**, *64*, 2545–2552. [[CrossRef](#)]
- Bartoszewicz, A.; Adamiak, K. Discrete Time Sliding Mode Control with a Desired Switching Variable Generator. *IEEE Trans. Autom. Control* **2020**, *65*, 1807–1814. [[CrossRef](#)]
- Saidi, S.; Abbassi, R.; Chebbi, S. Fuzzy Logic Controller for Three-level Shunt Active Filter Compensating Harmonics and Reactive Power. *Int. J. Adapt. Control Signal Process.* **2016**, *30*, 809–823. [[CrossRef](#)]
- Hornik, K. Multilayer Feedforward Networks are Universal Approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
- Man, Z.; Wu, H.; Palaniswami, M. An Adaptive Tracking Controller Using Neural Networks for a Class of Nonlinear Systems. *IEEE Trans. Neural Netw.* **1998**, *9*, 947–955.
- Liu, Z.; Liu, G.; Liu, J. Adaptive Tracking Controller using BP Neural Networks for A Class of Nonlinear Systems. *J. Syst. Eng. Electron.* **2004**, *15*, 598–604.
- Li, P.; Li, S. Learning Algorithm and Application of Quantum BP Neural Networks Based on Universal Quantum Gates. *J. Syst. Eng. Electron.* **2008**, *19*, 167–174.
- Schilling, R.; Carroll, J.J.; Al-Ajlouni, A.F. Approximation of Nonlinear Systems with Radial Basis Function Neural Networks. *IEEE Trans. Neural Netw.* **2001**, *12*, 1–15. [[CrossRef](#)]
- Li, Y.; Qiang, S.; Zhuang, X.; Kaynak, O. Robust and Adaptive Backstepping Control for Nonlinear Systems Using RBF Neural Networks. *IEEE Trans. Neural Netw.* **2004**, *15*, 693–701. [[CrossRef](#)] [[PubMed](#)]
- Lian, J.; Lee, Y.; Sudhoff, S.; Zak, S.H. Self-organizing Radial Basis Function Network for Real-time Approximation of Continuous-time Dynamical Systems. *IEEE Trans. Neural Netw.* **2008**, *19*, 460–474. [[CrossRef](#)] [[PubMed](#)]
- Sridhar, S.; Hassan, K.K. Output Feedback Control of Nonlinear Systems Using RBF Neural Networks. *IEEE Trans. Neural Netw.* **2000**, *11*, 69–79.
- Huang, G.; Saratchandran, P.; Sundararajan, N. A generalized growing and pruning RBF neural network for function approximation. *IEEE Trans. Neural Netw.* **2005**, *16*, 57–67. [[CrossRef](#)]
- Yang, C.; Chen, C.; He, W.; Cui, R.; Li, Z. Robot Learning System Based on Adaptive Neural Control and Dynamic Movement Primitives. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 777–787. [[CrossRef](#)] [[PubMed](#)]
- Chen, C.; Liu, Y.; Wen, G. Fuzzy Neural Network-based Adaptive Control for a Class of Uncertain Nonlinear Stochastic Systems. *IEEE Trans. Cybern.* **2014**, *44*, 583–593. [[CrossRef](#)]
- Wang, Z.; Fei, J. Fractional-Order Terminal Sliding Mode Control Using Self-Evolving Recurrent Chebyshev Fuzzy Neural Network for MEMS Gyroscope. *IEEE Trans. Fuzzy Syst.* **2021**. [[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. Li, S.; Wang, H.; Rafique, M.U. A Novel Recurrent Neural Network for Manipulator Control with Improved Noise Tolerance. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1908–1918. [[CrossRef](#)] [[PubMed](#)]
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.* **2021**, 1–11. [[CrossRef](#)]
24. Zhao, X.; Wang, X.; Zhang, S.; Zong, G. Adaptive Neural Backstepping Control Design for A Class of Non-smooth Nonlinear Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1820–1831. [[CrossRef](#)]
25. Fei, J.; Chen, Y.; Liu, L.; Fang, Y. Fuzzy Multiple Hidden Layer Recurrent Neural Control of Nonlinear System Using Terminal Sliding Mode Controller. *IEEE Trans. Cybern.* **2021**, 1–16. [[CrossRef](#)]
26. Fei, J.; Wang, Z.; Liang, X.; Feng, Z.; Xue, Y. Adaptive Fractional Sliding Mode Control of Micro gyroscope System Using Double Loop Recurrent Fuzzy Neural Network Structure. *IEEE Trans. Fuzzy Syst.* **2021**. [[CrossRef](#)]
27. Van, M.; Mavrovouniotis, M.; Ge, S. An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1448–1458. [[CrossRef](#)]
28. Hochreiter, S.; Jürgen, S. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
29. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [[CrossRef](#)]
30. Hou, B.; Zhou, Z. Learning with Interpretable Structure from Gated RNN. *IEEE Trans Neural Netw. Learn. Syst.* **2020**, *31*, 2267–2279. [[CrossRef](#)]
31. Zhang, S.; Yang, Y.; Xiao, J.; Liu, X.; Yang, Y.; Xie, D.; Zhuang, Y. Fusing Geometric Features for Skeleton-Based Action Recognition Using Multilayer LSTM Networks. *IEEE Trans. Multimed.* **2018**, *20*, 2330–2343. [[CrossRef](#)]
32. Yang, Y.; Zhou, J.; Ai, J.; Bin, Y.; Hanjalic, A.; Shen, H.T.; Ji, Y. Video Captioning by Adversarial LSTM. *IEEE Trans. Image Process.* **2018**, *27*, 5600–5611. [[CrossRef](#)] [[PubMed](#)]
33. Wang, P.; Jiang, A.; Liu, X.; Shang, J.; Zhang, L. LSTM-Based EEG Classification in Motor Imagery Tasks. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2018**, *26*, 2086–2095. [[CrossRef](#)] [[PubMed](#)]
34. Ergen, T.; Kozat, S. Efficient Online Learning Algorithms Based on LSTM Neural Networks. *IEEE Trans Neural Netw. Learn. Syst.* **2018**, *29*, 3772–3783.
35. Lippi, M.; Montemurro, M.; Esposti, M.; Cristadoro, G. Natural Language Statistical Features of LSTM-Generated Texts. *IEEE Trans Neural Netw. Learn. Syst.* **2019**, *30*, 3326–3337. [[CrossRef](#)]
36. Wang, Z.; Chai, J.; Xia, S. Combining Recurrent Neural Networks and Adversarial Training for Human Motion Synthesis and Control. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 14–28. [[CrossRef](#)]
37. Wang, H.; Luo, C.; Wang, X. Synchronization and Identification of Nonlinear Systems by Using a Novel Self-evolving Interval Type-2 Fuzzy LSTM-neural Network. *Eng. Appl. Artif. Intell.* **2019**, *81*, 79–93. [[CrossRef](#)]
38. Liu, L.; Fei, J. Novel Fuzzy Neural Network Sliding Mode Control of Active Power Filter. In Proceedings of the 2020 23rd International Conference on Electrical Machines and Systems (ICEMS), Hamamatsu, Japan, 24–27 November 2020.
39. Liu, L.; Fei, J.; An, C. Adaptive Sliding Mode Long Short-Term Memory Fuzzy Neural Control for Power Harmonic Suppression. *IEEE Access* **2021**, *9*, 69724–69734. [[CrossRef](#)]
40. 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](#)]