Article

Design of Adaptive TSK Fuzzy Self-Organizing Recurrent Cerebellar Model Articulation Controller for Chaotic Systems Control

Shun-Yuan Wang *, Chuan-Min Lin and Chen-Hao Li

Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan; t8319002@ntut.edu.tw (C.-M.L.); chenhaoli992@gmail.com (C.-H.L.)

* Correspondence: sywang@ntut.edu.tw

Abstract: The synchronization and control of chaos have been under extensive study by researchers in recent years. In this study, an adaptive Takagi–Sugeno–Kang (TSK) fuzzy self-organizing recurrent cerebellar model articulation controller (ATFSORC) is proposed, which is composed of a set of TSK fuzzy rules, a cerebellar model articulation controller (CMAC), a recurrent CMAC (RCMAC), a self-organizing CMAC (SOCMAC), and a compensation controller. Specifically, RCMAC, RCMAC, and adaptive laws are adopted so that the association memory layers of ATFSORC can be modulated in accordance with the layer decision-making mechanism in order to reduce the structure complexity and improve the control performance of ATFSORC. Moreover, the Takagi–Sugeno–Kang fuzzy rules are introduced to increase the learning speed of ATFSORC, and the improved compensating controller is designed to dispel the errors between an ideal controller and the TFSORC. Moreover, the proposed ATFSORC is applied to chaotic systems in order to validate its performance and feasibility. Several simulation schemes are demonstrated to show the effectiveness of the proposed method. Simulation results show that the proposed ATFSORC can obtain a favorable control performance when the chaotic systems are operated at different parameters. Specifically, ATFSORC can achieve faster convergence of the tracking error than fuzzy CMAC (FCMAC) and CMAC.

Keywords: cerebellar model articulation controller; self-organizing; recurrent; TSK fuzzy systems; chaotic systems

1. Introduction

Chaos is ubiquitous and exists in various scientific fields. Therefore, it has been receiving attention widely from many scholars and has been extensively studied [1–5]. The chaotic system is very sensitive to the initial conditions, so the behavior of a chaotic system is eventually unpredictable. If the chaotic phenomenon can be appropriately controlled, it may be of tremendous benefit in various applications because of the highly nonlinear characteristics of chaotic systems. The research overview for the synchronization and stability control of chaotic systems is discussed below.

Tao et al. proposed a type of synchronization control between gap junction-coupled neurons [6]. They used the advantage of the chattering-free feature in the adaptive sliding mode to perform chaotic synchronization. Hsu proposed a chaotic synchronization system based on a Takagi–Sugeno–Kang (TSK)-type Hermite neural network [7]. The control system consists of an artificial neural controller and a compensator, which provide good control performance. Lin et al. proposed a recurrent fuzzy cerebellar model articulation controller and used signed distance to convert dual input variables into single input variables, thereby reducing controller architectural complexity [8]. Che et al. proposed a robust adaptive artificial neural network controller for the synchronization of
two synchronous gap junction-coupled neurons in order to synchronize the chaotic system by appropriately selecting the control parameters [9]. Wu et al. proposed an adaptive TSK cerebellar model articulation controller for chaotic master–slave system synchronization by using TSK fuzzy rules and demonstrated good control tracking performance [10]. Wen et al. proposed a recurrent fuzzy cerebellar model articulation controller for dual-axis piezoelectric actuators with an input space quantization method and self-tuning learning rates [11]. This method allows the traditional cerebellar model articulation controller to have dynamic characteristics and ameliorate the shortcomings of a traditional cerebellar model, and it is effectively applied on uncertain systems. Lin et al. proposed a TSK fuzzy cerebellar model articulation controller for uncertain nonlinear systems as a tracking controller, in which the control system contains a TSK fuzzy cerebellar model articulation controller and a robust compensation controller; even if the system model is unknown, it can still achieve good tracking performance [12]. Lian et al. proposed an adaptive fuzzy self-organizing sliding-mode controller for active suspension systems by adding a sliding surface, and the stability could be more easily proven than by the traditional self-organizing fuzzy controller. Its model-free control design possesses online learning ability, which can be applied on any interfering complex system [13]. Lin et al. proposed a novel self-organizing wavelet cerebellar model articulation controller, which is applied on a nonlinear chaotic system [14]. By structure learning, it can correct the number of memory layers in real time to achieve the optimal wavelet cerebellar model articulation controller architecture, and by parameter learning, it can adjust the weight of the wavelet cerebellar model articulation controller to achieve the optimal approximation performance, and thus achieve a better control response. Chen proposed an architecture based on the TSK-type self-organizing recurrent artificial neural network that can generate or eliminate through self-organizing in order to obtain a network structure of an appropriate size [15]. Che et al. proposed an adaptive sliding-mode neural network and used a radial basis function to implement two gap junction-coupled neuron models for achieving chaotic synchronization [16].

In recent years, with the unceasing update and breakthrough of higher control theory, many outstanding scholars in the control field have successively proposed advanced control theories, such as the fuzzy control, artificial neural network (ANN), wavelet neural network (WNN), and self-organizing fuzzy control for nonlinear chaotic systems, switched-reluctance motor drive systems, voice coil motors, and permanent magnet synchronous motors. All have a fairly high reference value. Peng et al. proposed an adaptive recurrent cerebellar model articulation controller for linear ultrasonic motors, which facilitated dynamic performance and faster learning speed for the traditional cerebellar model articulation controller; they used the method of steepest descent to update the dynamic parameters of the controller in real time, thereby achieving the optimal learning velocity such that the tracking error is quickly converged [17]. Hung et al. proposed a wavelet fuzzy neural network that adopts an asymmetric membership function to ameliorate differential evolution algorithms, which are used to control an electric power steering system of a six-phase permanent magnet synchronous motor; it combines the advantages of the wavelet decomposition, fuzzy logic system, and asymmetric membership function [18]. It was developed to realize the control performance and stability required in electric power steering systems. Lin et al. proposed a novel adaptive wavelet fuzzy cerebellar model articulation control system for trajectory tracking control of voice coil motors; this control system includes a wavelet fuzzy cerebellar model articulation controller and a fuzzy compensation controller [19]. Since the wavelet function possessing differential characteristics and fuzzy are adopted, the controller has fast learning ability, due to which it shows significant performance and good tracking characteristics. Li et al. proposed a sliding-mode control method to be applied on a permanent magnet synchronous motor and added an interference estimator for feed-forward compensation; therefore, the controller possesses good robustness and can use a lower switching gain to ameliorate the chattering phenomenon of the sliding
mode [20]. Lin et al. proposed a dynamic Petri-fuzzy cerebellar model articulation controller, which was applied on a magnetic levitation system and a piezoelectric motor drive system; by adding the Petri network, the burden of parameter learning is reduced [20]. Lin et al. proposed a recurrent neural network based on differential learning algorithms and applied it to squirrel-cage induction generators, which can process nonlinear and non-differentiable functions through differential algorithms [20]. Lin et al. proposed a type of function-linked neural network using an asymmetric radial basis function for the three-dimensional motion control of a piezoelectric platform; by using an asymmetric radial basis function, the learning ability of the network can be improved [21]. Lin et al. proposed a type of TSK neural network controller using an asymmetric radial basis function; they used the architecture of functional fuzzy rules to improve the learning rate and applied it to a three-phase grid-connected photovoltaic system [22].

The aim of this research is to develop an adaptive TSK fuzzy self-organizing recurrent cerebellar model articulation controller (ATFSORC) for the synchronization and stability control of chaotic systems. By using ATFSORC, the master and slave systems that are seemingly in a chaotic state can be adjusted according to the reciprocal errors so that the slave system can be synchronized with the master system under different initial conditions. Moreover, TSK fuzzy rules are introduced into a recurrent cerebellar model articulation controller (RCMAC), which is combined with a self-organizing CMAC (SOCMAC) and an improved compensation controller to design the ATFSORC. By adopting the architecture combining SOCMAC, RCMAC, and adaptive laws, the cerebellar model articulation controller can possess dynamic memory performance and layer correction capability. Specifically, in the proposed controller, the number of association memory layers is modulated in accordance with the layer decision-making mechanism to reduce the architectural complexity of the ATFSORC and improve the control performance. The ATFSORC controller takes integrated error function as the input, which is then introduced into the TSK fuzzy self-organizing recurrent cerebellar model articulation controller, and an improved compensation controller is taken to compensate the errors between the ideal controller and TSK fuzzy self-organizing recurrent cerebellar model articulation controller. Moreover, this study adopted Lyapunov theorem to deduce the adaptive laws for the cerebellar model articulation controller’s weight value, TSK fuzzy rules’ parameter, recurrent weight value, Gaussian function midpoint, and Gaussian function standard deviation in order to ensure the stability of the system. Then, the proposed ATFSORC is applied to chaotic systems in order to validate its performance and feasibility. Table 1 shows the full forms and the corresponding abbreviations used in this study for better information.

Table 1. Full forms and the corresponding abbreviations used in this study.

| Full Form                                      | Abbreviation |
|-----------------------------------------------|--------------|
| Takagi–Sugeno–Kang                            | TSK          |
| Artificial Neural Network                     | ANN          |
| Wavelet Neural Network                        | WNN          |
| Cerebellar Model Articulation Controller      | CMAC         |
| Recurrent Cerebellar Model Articulation Control | RCMAC       |
| Self-Organizing Cerebellar Model Articulation Controller | SOCMAC |
| Fuzzy Cerebellar Model Articulation Controller | FCMAC       |
| Adaptive Takagi–Sugeno–Kang Fuzzy Self-Organizing Recurrent Cerebellar Model Articulation Controller | ATFSORC |

Appl. Sci. 2021, 11, 1567
2. Cerebellar Model Articulation Controller Theory

A cerebellar model articulation controller (CMAC) is a new neurological architecture proposed by Albus et al. in 1975 [23–26]. CMAC is a learning structure that mimics the human cerebellum. It is a neural network based on partial learning and can approximate nonlinear continuous functions with arbitrary precision. The CMAC neural network not only can hierarchically represent information but also can correct the partial weight coefficient. The learning velocity is high and the structure is simple. It possesses the advantages of local generalization, which makes it easier to be applied in real-time control compared with other neural networks; hence, it is one of the most widely applied neural networks in control.

As suggested by its name, the so-called cerebellar model articulation controller is derived by mimicking the operating manner of the human cerebellum. The architecture of CMAC is somewhat similar to an artificial neural network; in fact, it can be regarded as a simplified artificial neural network. The biggest difference is that unlike the artificial neural network, which uses only the triggered memory in each learning, CMAC will use all the memories in each learning, thereby drastically improving the learning velocity and usage of memory. Figure 1 is a schematic diagram of a generalized CMAC mapping. The architecture includes an input state space, an association memory space, a weight memory space, an actual output \(y\), a desired output \(y_d\), and a learning algorithm. Vector \(S\) of the input state, vector \(A\) of the association memory, and vector \(W\) of the weighted memory are, respectively, defined as follows:

\[
S = [s_1 \ s_2 \ \ldots \ s_n]^T \in \mathbb{R}^n, \tag{1}
\]

\[
A = [a_1 \ a_2 \ \ldots \ a_p]^T \in \mathbb{R}^p, \tag{2}
\]

\[
W = [w_1 \ w_2 \ \ldots \ w_p]^T \in \mathbb{R}^p. \tag{3}
\]

![Figure 1. Schematic diagram of cerebellar model articulation controller mapping.](image)

1. Quantization of continuous inputs:

At the beginning of the CMAC action, the quantization step is first performed to quantize the input learning sample into \(n\) discrete input states \((s_1 \sim s_n)\) in order to delimit each input state to a limited closed interval for variation. A higher number of quantization implies a higher learning resolution. Although the arithmetic precision can be increased, thereby increasing the control precision, it consumes relatively more memories; thereby, the arithmetic velocity is reduced, and thus the learning velocity decreases. Therefore, control precision and learning velocity are subjected to a trade-off for the user’s needs.
2. Mapping of input state space to association memory space:
After the input signal goes through the CMAC quantization step, the quantized input state is mapped to the association memory space. For mapping the input state space to the association memory space, an input state must be mapped to at least two or more association memory addresses, and the association memory addresses mapped by the adjacent input state need to have at least one overlap. The reason why CMAC has to satisfy the aforementioned rules in the input state space mapping to association memory space is mainly that CMAC possesses the ability to infer their similar outputs of unlearned samples through the adjacent memory weights that have been learnt and corrected, i.e., the ability of local generalization.

3. Mapping of association memory space to weight memory space:
The aim of association memory mapping to weight memory is to correspond the association memory reference state to relevant weight memory. The association memory element in the association memory space is composed of 0 and 1. In simple words, the fired memory addresses are 1, and the remaining memory addresses that are not fired are 0.

4. Weight memory space to output space:
When the mapping process is completed, the inner products of vector $A$ of association memory and vector $W$ of weight memory are taken as the output values of CMAC, as shown in Equation (4). This is the recall of CMAC, and the weight value will be adjusted and corrected on the basis of the learning algorithm such that the actual output $y$ in the next CMAC learning can be closer to the desired output $y_u$.

$$S = [s_1 \ s_2 \ ... \ s_n]^T \in \mathbb{R}^n,$$

where $y$ is the actual output; $A$ is the association memory vector; $W$ is the weight memory vector; $p$ is the number of memory layers; $a_i$ is the $j$-th association memory address; and $w_j$ is the $j$-th weight.

5. Weight update:
The learning algorithm of CMAC is as shown in Equation (5). It is mainly the error obtained by subtracting the actual output from the desired output. Based on the learning algorithm, errors are fed back to the fired weight in an evenly distributed manner to generate a weight correction amount and correct the original weight value. Training is continuously performed until the output error converges with an allowable range.

$$w_j(k) = w_j(k-1) + \Delta w_j(k)$$

$$= w_j(k-1) + \frac{\eta}{P} a_j(y_j(k) - y(k)).$$

The magnitude of learning rate $\eta$ will greatly influence the learning process. If the learning rate is set to be large, the learning velocity will increase, and the system response will be faster; however, the resulting learning process will considerably change, which leads to system instability. If the learning rate is set to be small, the learning time will be longer, leading to a reduced convergence velocity of the system, and thus the system cannot reach the designated range of convergence in a short time.

3. Adaptive TSK Fuzzy Self-Organizing Recurrent Cerebellar Model Articulation Controller Design

3.1. TSK Fuzzy System
TSK fuzzy rules, also known as functional fuzzy rules, are extended from the Mamdani fuzzy rules. Compared with Mamdani fuzzy rules, there are the following characteristics. In addition, the fuzzy membership function of input $s$ is shown in Figure 2.
The TSK fuzzy rules (1) possess high computational efficiency and are suitable for systems with high real-time requirements; (2) they can be combined very well with linear systems, optimized control, and adaptive control systems; and (3) they can guarantee the smoothness of controller outputs.

The difference between TSK fuzzy rules and Mamdani fuzzy rules is that the posterior part of TSK fuzzy rules is a linear combination of input states using the coefficients for linear combination of input states from the online adjusted fuzzy posterior part. The TSK fuzzy rules are as follows:

\[
R^{(1)}: \text{if } s \text{ is NB, then } y_1 = C_0^1 + C_1^1 s, \\
R^{(2)}: \text{if } s \text{ is NM, then } y_2 = C_0^2 + C_1^2 s, \\
R^{(3)}: \text{if } s \text{ is NS, then } y_3 = C_0^3 + C_1^3 s, \\
R^{(4)}: \text{if } s \text{ is Z, then } y_4 = C_0^4 + C_1^4 s, \\
R^{(5)}: \text{if } s \text{ is PS, then } y_5 = C_0^5 + C_1^5 s, \\
R^{(6)}: \text{if } s \text{ is PM, then } y_6 = C_0^6 + C_1^6 s, \\
R^{(7)}: \text{if } s \text{ is PB, then } y_7 = C_0^7 + C_1^7 s.
\]

If the fuzzy inference adopts the product inference engine, the defuzzification will adopt the center average method. The output is as shown in Equation (6).

\[
u = \frac{\sum_{i=1}^{7} v_i y_i}{\sum_{i=1}^{7} v_i} = \frac{\sum_{i=1}^{7} v_i [C_0^i, C_1^i] [1, s]}{\sum_{i=1}^{7} v_i} = \Phi^T \Psi,
\]

where

\[
\Phi^T = [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7], \\
\phi_i = [C_0^i, C_1^i], \\
\Psi = [\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \psi_7]^T, \\
\psi_i = \frac{1}{\sum_{i=1}^{7} v_i} [1, s], \\
v_i \text{ is the degree of membership for the input membership function,} \\
y_i \text{ is the linear combination outputs of the posterior part.}
\]

3.2. Design of TSK Fuzzy Self-Organizing Recurrent Cerebellar Model Articulation Controller

When CMAC is applied, the number of layers for association memory is fixed. Depending on the input state, the association memory value is 1 when being triggered; when not being triggered, it will be 0. However, such a binary relationship will lower the network resolution and cause the problem of control outputs that are not smooth. This study proposes a self-organizing recurrent cerebellar model articulation controller; the architecture is shown in Figure 3. With the working principle similar to that of a
traditional CMAC, the association memory layer is implanted with Gaussian functions to cover the entire input space. Unlike the traditional rectangular function, the Gaussian function possesses the characteristics of being differentiable at any position; therefore, not only can the controller perform function learning but also the resulting outputs can have differential properties. Moreover, similar self-organizing recurrent artificial neural network architecture and adaptive rules are adopted so that the CMAC that is originally static with a fixed number of association memory layers possesses dynamic memory ability and the number of association memory layers can be changed along with the layer decision-making mechanism.

![Diagram](image)

**Figure 3.** Takagi–Sugeno–Kang (TSK) fuzzy self-organizing cerebellar model articulation controller architecture.

As shown in Figure 3, the proposed architecture contains an input space $S$, an association memory space $A$, a receptive-field space $B$, a function coupling output space $F$, real memory space $Z$, and output space $Y$. The layer decision-making mechanism of association memory is shown in Figure 4.

![Diagram](image)

**Figure 4.** Layer decision-making mechanism.

The vector of input signals is as shown in Equation (7). It will first be subjected to quantification before entering the TSK fuzzy self-organizing recurrent cerebellar model articulation controller, which takes the range $[-1, 1]$ as the input.
\[ s = [s_1, s_2, \ldots, s_n]^T \in \mathbb{R}^n. \] (7)

Elements in each input variable vector will be added to the product of the previous association memory output and the recurrent weight, called the recurrent input vector \( \mathbf{x} \), as shown in Equations (8) and (9).

\[
\mathbf{x} = [x_1, x_2, \ldots, x_p]^T \in \mathbb{R}^p, \tag{8}
\]

\[
x_j(k) = s_j(k) + r_j a_j(k - 1) \text{ for } j = 1, 2, \ldots, p, \tag{9}
\]

where \( n \) is the individual number of input states; \( p \) is the number of layers in the association memory space; \( k \) is the \( k \)-th sampling time; and \( r_j \) is the recurrent weight value. Each variable will correspond to a set of association memories. The pre-set number of layers in the association memory is one layer, and each layer is implanted with one Gaussian function. When the input variable is mapped to each Gaussian function, different degrees of membership \( a_j \) will be generated, as shown in Equation (10).

\[
a_j = \exp \left[ -\frac{(x_j - m_j)^2}{v_j^2} \right] \text{ for } j = 1, 2, \ldots, p \tag{10}
\]

where \( x_j \) is the recurrent input; \( m_j \) is the midpoint of the Gaussian function; and \( v_j \) is the standard deviation of the Gaussian function. The membership level \( a_j \) generated from the association memory layer is individually and correspondingly multiplied and mapped into the receptive field vector \( \mathbf{b} \), and the definition of the receptive field vector is shown in Equation (11).

\[
b_j = a_j \text{ for } j = 1, 2, \ldots, p, \tag{11}
\]

This is expressed in a vector as

\[
\mathbf{b}(\mathbf{x}, \mathbf{m}, \mathbf{v}, \mathbf{r}) = [b_1, b_2, \ldots, b_p]^T,
\]

where \( \mathbf{v} = [v_1, v_2, \ldots, v_p]^T \), \( \mathbf{m} = [m_1, m_2, \ldots, m_p]^T \), and \( \mathbf{r} = [r_1, r_2, \ldots, r_p]^T \). In this study, the physical memory space is the product of the receptive field space and the TSK output; each recurrent input state \( x_j \) corresponds to a TSK fuzzy system. Wherein, the fuzzy control rules for the \( j \)-th fuzzy system are designed as follows:

- \( R^{(1)} \): if \( x_j \) is \( \text{NB} \), then \( t_{j1} = c_{j0}^1 + c_{j1} x_j \),
- \( R^{(2)} \): if \( x_j \) is \( \text{NM} \), then \( t_{j2} = c_{j0}^2 + c_{j1} x_j \),
- \( R^{(3)} \): if \( x_j \) is \( \text{NS} \), then \( t_{j3} = c_{j0}^3 + c_{j1} x_j \),
- \( R^{(4)} \): if \( x_j \) is \( \text{Z} \), then \( t_{j4} = c_{j0}^4 + c_{j1} x_j \),
- \( R^{(5)} \): if \( x_j \) is \( \text{PS} \), then \( t_{j5} = c_{j0}^5 + c_{j1} x_j \),
- \( R^{(6)} \): if \( x_j \) is \( \text{PM} \), then \( t_{j6} = c_{j0}^6 + c_{j1} x_j \),
- \( R^{(7)} \): if \( x_j \) is \( \text{PB} \), then \( t_{j7} = c_{j0}^7 + c_{j1} x_j \),

where \( b_n \) is the linear combination of the posterior parts in TSK, as shown in Equation (13).

\[
t_{jq} = c_{j0}^q + c_{j1} x_j = f_j^q, \quad q = 1, 2, \ldots, 7, \tag{13}
\]

where \( f_j = [1, x_j]^T \), \( f = [1, x_j]^T \), for \( j = 1, 2, \ldots, p \). The function value in receptive field vector \( \mathbf{b} \) is defined as firing strength \( \alpha_j \), which is used for determining the self-organizing layer decision-making mechanism.

\[
\alpha_j = b_j, \quad \text{for } j = 1, 2, \ldots, p. \tag{14}
\]
Next, it is to determine the maximum value \( \alpha_{\text{max}} \) from these firing strengths. The maximum firing strength \( \alpha_{\text{max}} \) is compared with a preset threshold value \( \alpha_{\text{th}} \). If \( \alpha_{\text{max}} \) is less than or equal to \( \alpha_{\text{th}} \), one new layer of association memory will be added. The threshold value \( \alpha_{\text{th}} \) is in between 0 and 1.

\[
\alpha_{\text{max}} = \max_{j=1}^{p(k)} \alpha_j, \tag{15}
\]

\[
\alpha_{\text{max}} \leq \alpha_{\text{th}}. \tag{16}
\]

The determinant of Equation (16) will cause the number of layers to increase by at most one layer at a time, as shown in Equation (17). Newly added association memory is equally implanted with one Gaussian function. For the new Gaussian function, the midpoint \( m_j^{\text{new}} \), standard deviation \( \sigma_j^{\text{new}} \), weight \( w_{ij}^{\text{new}} \), and recurrent weight value \( r_j^{\text{new}} \) are defined as below:

\[
p(k + 1) = p(k) + 1, \tag{17}
\]

\[
m_{p+1}^{\text{new}} = x_j, \tag{18}
\]

\[
\sigma_{p+1}^{\text{new}} = \sigma_0, \tag{19}
\]

\[
w_{p+1}^{\text{new}} = w_0, \tag{20}
\]

\[
r_{p+1}^{\text{new}} = r_0. \tag{21}
\]

Here, \( x_j \) is the current input state; \( \sigma_0 \) and \( r_0 \) are the constants given in advance; and \( w_0 \) is the TSK parameter given in advance. If \( \alpha_j \) is less than or equal to the threshold value \( \beta_{\text{th}} \), the contribution of the layer will not be large, so deleting that layer of association memory will not affect the control performance; thus, the computation amount can be reduced.

\[
\alpha_j \leq \beta_{\text{th}}. \tag{22}
\]

The physical memory space is shown by Equation (23).

\[
z(x, m, v, r, c, f) = l_j b_j, \tag{23}
\]

where \( j = 1, 2, \ldots, p \) and \( q = 1, 2, \ldots, 7 \). After summation, the resulting total output control amount is as shown in Equation (24).

\[
y = u_{\text{TFSORC}} = \sum_{j=1}^{p} z_j = \sum_{j=1}^{p} t_{jq} b_j = \sum_{j=1}^{p} e_j \Lambda b_j = W^T \Lambda, \tag{24}
\]

where \( \Lambda = [b_1^T, b_2^T, \ldots, b_7^T]^T; W = [c_1^T, c_2^T, \ldots, c_7^T]^T. \)

The proposed TFSORC can enable the CMAC with originally static and fixed architecture to possess dynamic memory performance, and the number of association memory layers can increasingly or decreasingly change along with the self-organizing layer decision-making mechanism in order to improve the controller output precision and reduce the computation amount, thereby providing better control performance than the traditional CMAC.

3.3. Design of Adaptive TSK Fuzzy Self-Organizing Recurrent Cerebellar Model Articulation Controller for Chaotic Synchronization Control System

In this study, TFSORC combines the adaptive theorem with an integrated error function along with an improved compensation controller to form the adaptive TSK fuzzy self-organizing recurrent cerebellar model articulation controller (ATFSORC). It takes the signals of integrated error function as the input, as shown in Figure 5. Here, \( y_d \) is the desired output; \( y \) is the actual output; \( e \) is the difference value between the desired output and actual output; \( \beta_v, \beta_w, \beta_r \), and \( \beta_d \) are the learning rates of weight value for weight
memory, midpoint, and standard deviation for Gaussian function, and recurrent weight value, respectively; and $\gamma$ is the compensation coefficient. The total control amount $u$ of ATFSORC is as shown in Equation (25), where $u_{TFSORC}$ is the control amount of the TSK fuzzy self-organizing cerebellar model articulation controller; and $u_c$ is the control amount of the improved compensation controller. When there is an ideal control amount $u^*$, the error between the $u_{TFSORC}$ of the TSK fuzzy self-organizing cerebellar model articulation controller and the ideal control amount $u^*$ will be compensated by the improved compensation controller $u_c$.

$$u = u_{TFSORC} + u_c.$$  \hspace{2cm} (25)

![Figure 5. Adaptive TSK fuzzy self-organizing cerebellar model articulation controller architecture.](image)

In this study, the proposed ATFSORC is used as a synchronization controller for chaotic systems. As shown in Figure 6, the controlled body in Figure 5 is converted into a slave system; $y_\mu$ can be regarded as a master system output ($x_1, y_1$); $y$ can be regarded as a slave system output ($x_2, y_2$); $e_x$ and $e_y$ are the error values of $x_1, x_2$ and $y_1, y_2$, respectively. Moreover, in order to reduce the architectural complexity of ATFSORC, a method of signed distance is introduced to convert the two errors $e_x$ and $e_y$ in Figure 6 into an error $v$. Figure 7 shows the $e_x - e_y$ plane and the switch line $L_v$, which is as shown in Equation (26).

$$L_v : \lambda e_x + e_y = 0,$$  \hspace{2cm} (26)

where $\lambda$ is a normal number; and $v$ is the signed distance between $M(e_x, e_y)$ and $L_v$, as shown in Equation (27).

$$v = (\lambda e_x + e_y) / \sqrt{1 + \lambda^2}.$$  \hspace{2cm} (27)
In a neuronal system, a gap junction is an electrical synapse. Through the gap junction, neurons can communicate with each other, and the synaptic current is directly proportional to the membrane potentials between the two neurons. In the previous literature [26], the effect of gap junction on the synchronization of two coupled FitzHugh–Nagumo (FHN) neurons was studied. A circuit diagram of the neuron model is shown in Figure 8, where:

- $V_1, V_2$: Membrane voltages of two gap-junction-coupled neurons,
- $f(V_1), f(V_2), L_m, C_m$: Electrical parameters of neurons,
- $g$: Coupling strength between neurons,
- $I_o$: External electrical stimulation.

The two gap-junction-coupled FHN neuron model is expressed in Equations (28) and (29).

Master system:
\[ \dot{x}_i = x_i(x_i - 1)(1 - rx_i) - y_i - g(x_i - x_2) + I. \]  
\[ \dot{y}_i = bx_i. \]  
\[ \dot{x}_2 = x_2(x_2 - 1)(1 - rx_2) - y_2 - g(x_2 - x_1) + I + u. \]  
\[ \dot{y}_2 = bx_2. \]

Slave system:

First, the errors of two gap-junction-coupled FHN neurons are defined as follows:

\[ e_x = x_1 - x_2 \]
\[ e_y = y_1 - y_2. \]  

Next, the error dynamic system can be expressed as Equation (31).

\[ \dot{e}_x = \dot{x}_1 - \dot{x}_2 \]
\[ = -rx_1^3 + (r + 1)x_1^2 + rx_2^3 - (r + 1)x_2^2 \]
\[ - 2g(x_1 - x_2) - (y_1 - y_2) - u. \]  
\[ \dot{e}_y = \dot{y}_1 - \dot{y}_2 \]
\[ = b(x_1 - x_2) \]

Merging the two errors of Equation (30) through signed distance into one error \( v \), as shown in Equation (32), we get

\[ v = (\lambda_1 e_x + e_y) / \sqrt{1 + \lambda_1^2}. \]

Differentiating Equation (32) and substituting in Equation (31) gives Equation (33).

\[ \dot{v} = (\lambda_2 \dot{e}_x + \dot{e}_y) / \sqrt{1 + \lambda_2^2} \]
\[ = -2g[\lambda_1(x_1 - x_2) + (y_1 - y_2)][\sqrt{1 + \lambda_1^2} - \lambda_2u / \sqrt{1 + \lambda_2^2}] \]
\[ + (2\lambda_1 - \lambda_1)[(y_1 - y_2) + (b - \lambda_1)(x_1 - x_2)]/\sqrt{1 + \lambda_1^2} \]
\[ = A v + B u + d \]  

where

\[ A = -2g, \]
\[ B = \frac{-\lambda_1}{\sqrt{1 + \lambda_1^2}}, \]
\[ d = (2\lambda_1 - \lambda_1)[(y_1 - y_2) + (b - \lambda_1)(x_1 - x_2)]/\sqrt{1 + \lambda_1^2}. \]

In the actual chaos, considering the dynamics of a chaotic system, the parameters will change accordingly under different conditions. Therefore, Equation (32) can be rewritten as

\[ \dot{v} = (A_n + \Delta A)v + (B_n + \Delta B)u + (d + \Delta d) \]
\[ = A_n v + B_n u + \Delta A v + \Delta B u + d_1 \]  
\[ = A_n v + B_n u + d_2 \]  

where \( A_n \) and \( B_n \) are the nominal values of \( A \) and \( B \), respectively; \( \Delta A \) and \( \Delta B \) are the variation values of \( A \) and \( B \), respectively; and \( d_1 \) and \( d_2 \) are the uncertainty dynamic model (system uncertainty), in which \( d_1 = d + \Delta d \), \( d_2 = \Delta A v + \Delta B u + d_1 \). Then, transposing and arranging the equation, we can obtain

\[ u = \frac{1}{B_n}[-A_n v - d_2 + \dot{v}]. \]  

Ideally, if all the parameters of a chaotic system are known, an ideal control amount \( u^* \) can be constructed.
\[ u^* = \frac{1}{B_x} \left[ -A_x v - d_z + \dot{v}^* - k_e e \right], \]  

(36)

where \( v^* \) is the ideal error amount; and \( k_e \) is a normal number. Error \( e \) is defined as Equation (37).

\[ e = v - v^*. \]  

(37)

If the control amount \( u \) approaches the ideal control amount \( u^* \), Equation (38) can be obtained based on Equations (35) and (36); Equation (39) can be obtained from Equations (37) and (38).

\[ \dot{v} = \dot{v}^* - k_e e, \]  

(38)

\[ \dot{e} + k_e e = 0. \]  

(39)

Since \( k_e \) is a positive real number and Equation (39) is a strictly Hurwitz polynomial, the error will converge to zero as time passes to infinity, i.e., \( \lim_{t \to \infty} e = 0 \). Next, the integrated error function \( s_i \) is defined as in Equation (40).

\[ s_i(t) = e(t) + k_e \int_0^t e(\tau) d\tau. \]  

(40)

Differentiating Equation (40) with respect to time \( t \), we get

\[ \dot{s}_i = \dot{e} + k_e e. \]  

(41)

When the control amount is the ideal control amount \( u^* \), the error \( e \) can become asymptotically stable. However, in fact, such an ideal control amount cannot be obtained because the system parameters and an unknown dynamic model will vary along with many uncertain factors, and these cannot be known precisely in advance. Therefore, this study proposes ATFSORC, which can use the good approximation capability of a Takagi–Sugeno–Kang (TSK) fuzzy self-organizing recurrent cerebellar model articulation controller (TFSORC) to learn the ideal control amount \( u^* \) and use the compensation controller to compensate for the error between TFSORC and \( u^* \). Figure 6 shows the ATFSORC architecture diagram for chaotic system synchronization. Its output amount is

\[ u = u_{TFSORC} + u_c, \]  

(42)

where \( u \) is the total control amount; \( u_{TFSORC} \) is the output amount of TFSORC; and \( u_c \) is the output amount of compensation controller.

Next, the association memory space is partitioned such that the optimal number of stage layers is \( p^* \), which is divided into an activated part as \( p \) and an inactivated part as \( p^* - p \). Therefore, the optimal parameters \( \Lambda^*, W^*, m^*, v^* \), and \( r^* \) are also divided into two parts.

\[ \Lambda^* = \begin{bmatrix} \Lambda_x^* \\ \Lambda_z^* \end{bmatrix}, \]  

(43)

\[ W^* = \begin{bmatrix} W_x^* \\ W_z^* \end{bmatrix}, \]  

(44)

\[ m^* = \begin{bmatrix} m_x^* \\ m_z^* \end{bmatrix}, \]  

(45)

\[ v^* = \begin{bmatrix} v_x^* \\ v_z^* \end{bmatrix}, \]  

(46)

\[ v^* = \begin{bmatrix} v_x^* \\ v_z^* \end{bmatrix}, \]  

(47)
Here, \( \Lambda^*, W^*, m^*, v^*, \text{ and } r^* \) are the optimal parameters relative to \( \Lambda, W, m, v, \text{ and } r \), respectively; \( \Lambda_o, W_o, m_v, v_o, \text{ and } r \) represent the activated parts, whereas \( \Lambda_o, W_o, m_v, v_o, \text{ and } r \) represent the inactivated parts.

Based on the universal approximation theorem, it is assumed that there is an optimal output control amount for the TSK fuzzy self-organizing recurrent cerebellar model articulation controller as \( u^*_{\text{TSORC}} \), and the ideal control amount \( u^* \) is expressed as Equation (48).

\[
\begin{align*}
  u^* &= u^*_{\text{TSORC}}(x, W^*, m^*, v^*, r^*) + \rho \equiv W^{rT} \Lambda^* + \rho, \\
  &\quad = W^*_a \Lambda^*_a + W^*_b \Lambda^*_b + \rho 
\end{align*}
\]

(48)

where \( \rho \) represents the error between the optimal and ideal controller amount. Since the optimal output control amount \( u^*_{\text{TSORC}} \) cannot be obtained easily, the \( u^*_{\text{TSORC}} \) value is estimated in an iterative manner. That is, the following is assigned:

\[
\begin{align*}
  u &= u^*_{\text{TSORC}}(x, \hat{W}_a, \hat{m}_a, \hat{v}_a, \hat{r}_a) + u_c \equiv \hat{W}_a^T \hat{\Lambda}_a + u_c, \\
  \hat{\Lambda}_a &= \Lambda^+_a - \hat{\Lambda}_a; \quad \hat{W}_a = W^*_a - \hat{W}_a.
\end{align*}
\]

(49)

By substituting Equations (48) and (49) into the formula, Equation (50) can be rewritten as

\[
\dot{s}_x = \hat{e} + k_se = B_e[u^* - u].
\]

(50)

Through Taylor series, the following expansions can be obtained.

\[
\begin{align*}
  &\tilde{\Lambda}_a = \Lambda^*_a - \hat{\Lambda}_a = \\
  &\begin{bmatrix}
  \hat{\Lambda}_1 \\
  \hat{\Lambda}_2 \\
  \vdots \\
  \hat{\Lambda}_r
\end{bmatrix} = \\
  &\begin{bmatrix}
  \frac{\partial h_f^T}{\partial m_a} \\
  \frac{\partial h_f^T}{\partial v_a} \\
  \vdots \\
  \frac{\partial h_f^T}{\partial r_a}
\end{bmatrix}_{m = \hat{m}_a, v = \hat{v}_a, r = \hat{r}_a} + \\
  &\begin{bmatrix}
  \frac{\partial h_f^T}{\partial m_a} \\
  \frac{\partial h_f^T}{\partial v_a} \\
  \vdots \\
  \frac{\partial h_f^T}{\partial r_a}
\end{bmatrix}_{m = m^*_a, v = v^*_a, r = r^*_a} + (v^*_a - \hat{v}_a) + (r^*_a - \hat{r}_a) + O,
\end{align*}
\]

(52)

where \( O \) is a vector of higher-order components. For convenience of derivation, assigning

\[
\begin{align*}
  \hat{m}_a &= m^*_a - \hat{m}_a, \quad \hat{v}_a = v^*_a - \hat{v}_a, \quad \hat{r}_a = r^*_a - \hat{r}_a \text{ and} \\
  &C = \begin{bmatrix}
  \frac{\partial h_f^T}{\partial m_a}, \frac{\partial h_f^T}{\partial v_a}, \ldots, \frac{\partial h_f^T}{\partial m_a}
\end{bmatrix}
\end{align*}
\]

(53)
Theorem 1. Considering a chaotic synchronization control system as Equation (34), if the synchronization controller is designed to be Equation (49) and while the ATFSORC’s adaptive laws and compensation controller output are as shown from Equation (60) to Equation (64), then the stability of such a controlled system can be guaranteed.

Proof of Theorem 1. Using the Lyapunov theorem to derive the adaptive laws, the Lyapunov function candidate \( V_\alpha \) is first defined as follows:

\[
V_\alpha = \frac{1}{2}v_\alpha^T v_\alpha + B_\alpha \left[ \frac{1}{2} w_\alpha^T w_\alpha + \frac{1}{2} m_\alpha^T m_\alpha \right] + \frac{1}{2} \tilde{v}_\alpha^T \tilde{v}_\alpha + \frac{1}{2} \tilde{r}_\alpha^T \tilde{r}_\alpha
\]

where \( \beta_{\alpha}, \beta_{m\alpha}, \beta_{w\alpha}, \) and \( \beta_{\alpha} \) are normal numbers greater than zero. Differentiating Equation (65) and substituting Equation (58) into the equation can give
\[
\dot{V}_c = s_a \dot{s}_a - B_stVcW_a + \frac{1}{\beta_{st}} \tilde{m}_a \dot{\tilde{m}}_a + \frac{1}{\beta_{st}} \tilde{v}_a \dot{\tilde{v}}_a + \frac{1}{\beta_{st}} \tilde{r}_a \dot{\tilde{r}}_a \\
= s_a B_s \left( \dot{W}_a^T Z_a + \dot{W}_a^T \left( C^T \tilde{m}_a + G^T \tilde{v}_a + H^T \tilde{r}_a \right) + D_s - u_s \right) \\
- \frac{B_s}{\beta_{st}} \dot{W}_a^T \dot{W}_a + \frac{1}{\beta_{st}} \dot{\tilde{m}}_a \dot{\tilde{m}}_a + \frac{1}{\beta_{st}} \dot{\tilde{v}}_a \dot{\tilde{v}}_a + \frac{1}{\beta_{st}} \dot{\tilde{r}}_a \dot{\tilde{r}}_a \\
= s_a B_s \left( \dot{W}_a^T \dot{Z}_a + s_s B_s \dot{W}_a^T C \tilde{m}_a + s_s B_s \dot{W}_a^T G^T \tilde{v}_a + s_s B_s \dot{W}_a^T H^T \tilde{r}_a \right) + \dot{\bar{V}}_a + \tilde{m}_a \left( s_s B_s \dot{W}_a^T C - \frac{B_s}{\beta_{st}} \tilde{m}_a \right) \\
+ \tilde{v}_a \left( s_s B_s \dot{W}_a^T G^T - \frac{B_s}{\beta_{st}} \tilde{v}_a \right) + \tilde{r}_a \left( s_s B_s \dot{W}_a^T H^T - \frac{B_s}{\beta_{st}} \tilde{r}_a \right) \\
(66)
\]

As \( V_c \geq 0 \), and in order to achieve the goal of \( \dot{V}_c < 0 \), we can assign
\[
\tilde{W}_a \left[ s_s B_s \hat{Z}_a - \frac{B_s}{\beta_{st}} \hat{W}_a \right] = 0, \\
(67)
\]
\[
\left[ s_s B_s \hat{W}_a C - \frac{B_s}{\beta_{st}} \hat{m}_a \right] \tilde{m}_a = 0, \\
(68)
\]
\[
\left[ s_s B_s \hat{W}_a G^T - \frac{B_s}{\beta_{st}} \hat{v}_a \right] \tilde{v}_a = 0, \\
(69)
\]
\[
\left[ s_s B_s \hat{W}_a H^T - \frac{B_s}{\beta_{st}} \hat{r}_a \right] \tilde{r}_a = 0. \\
(70)
\]

Substituting all formulas from Equation (60) to Equation (64) into Equation (66), we get
\[
\dot{V}_c = s_a B_s \left( D_s - \delta_s \text{sgn}[s_a] \right) \\
= s_a B_s D_s - \delta_s |B_s| |s_a| \leq -\delta_s |B_s| |s_a| \\
(71)
\]

From Equation (71), as \( \dot{V}_c \leq 0 \) is a negative semi-definite function, there is no guarantee that the error will converge to zero. By using Barbalat’s lemma, we can prove that when time approaches infinity, \( \dot{V}_c(t) \) will approach zero.

The function of \( \Xi_i(t) \) is defined as follows:
\[
\Xi_i(t) = (\delta_i - |D_s| |B_s| |s_a|) \leq -\dot{V}_c(t). \\
(72)
\]

Integrating Equation (72) with respect to time can give Equation (73), where \( V_c(t) \) is as shown in Equation (65).
\[
\int_0^t \Xi_i(\tau) d\tau \leq V_c(0) - V_c(t) \\
(73)
\]

Since \( V_c(0) \) is bounded, while \( V_c(t) \) is bounded and a non-incremental function, it can be determined from Equation (73) that \( \Xi_i(t) \) is bounded by time, i.e.,
\[
\lim_{t \to \infty} \int_0^t \Xi_i(\tau) d\tau < \infty. \\
(74)
\]

Next, since \( V_c(t) > 0 \) and \( \dot{V}_c(t) \leq 0 \), Equation (65) shows that \( \tilde{W}_a, \tilde{m}_a, \tilde{v}_a, \text{ and } \tilde{r}_a \) are bounded. As inferred from Equation (71), when \( s_a \neq 0 \), Equation (58) implies that \( \dot{s}_a \) is bounded, and \( s_a \) is uniformly continuous; therefore, from Equation (72), it is learnt that \( \Xi_i(t) \) is bounded, thus proving that \( \dot{V}_c(t) \) is uniformly continuous and \( \dot{V}_c(t) \) is bounded. From Barbalat’s Lemma, we can further obtain the following:
\[
\lim_{t \to \infty} \dot{V}_c(t) = 0, \forall s_a \neq 0. \\
(75)
\]
In addition, from Equation (71), when \( s_a = 0 \), \( \dot{y}_d(t) = 0 \). The above-mentioned inference indicates that for arbitrary \( s_a \) when the time \( t \) approaches infinity, \( \dot{y}_d(t) \) will converge to zero, and \( s_a \) at this point will also converge to zero. Equation (41) shows that if an appropriate \( k_1 \) is selected to conform to the Hurwitz polynomial, the errors \( e \) and \( v \) will also converge to zero when the time approaches infinity. Through the above derivations, we can ensure that the designed adaptive TSK fuzzy self-organizing recurrent cerebellar model articulation controller system is stable. □

3.4. Design of Improved Compensator

The adaptive TSK fuzzy self-organizing cerebellar model articulation controller proposed in this study contains an improved compensation controller. Equation (64) indicates that the output of compensation controller \( u_c \) is composed of a normal number \( \delta_i \) and a sign function of \( s_a \), wherein the function of \( \text{sgn}[s_a] \) is defined as in Equation (76). Since \( \delta_i \) is a normal number, the output of the compensation controller depends on the integrated error function \( s_a \) value. In fact, there are only three types of states as \( \delta_i \), 0, and \( -\delta_i \), which cause the shortcoming of having insufficient transient-state compensation or excessive steady-state compensation.

\[
\text{sgn}[s_a] = \begin{cases} 
1 & s_a > 0 \\
0 & s_a = 0 \\
-1 & s_a < 0 
\end{cases}
\]

(76)

Therefore, this study proposes an improved compensator to ameliorate the original compensation controller as presented arithmetically in Equation (64). The design of an improved compensation controller focuses on allowing the compensation controller to provide a larger compensation amount in ATFSORC at the transient state and a smaller compensation amount after ATFSORC enters the steady state.

Errors will increase when the compensation amount provided by the compensation controller is insufficient; therefore, the integrated error function \( s_a \) can be used as the basis to determine whether the ATFSORC has completed learning and entered the steady state. The above-mentioned design concept can be implemented from Equation (77).

\[
u_c = \begin{cases} 
\delta_1 \gamma_1 s_a & |s_a| > \zeta_1 \\
\delta_1 \text{sgn}[s_a] & 0 < |s_a| < \zeta_1
\end{cases}
\]

(77)

where \( \zeta_1 \) is the threshold value for switching the compensation control amount; \( \gamma_1 \) is a compensation coefficient; and \( \gamma_1 > 1 \), which is used to elastically adjust the magnitude of compensation amount in the transient state. When \( |s_a| > \zeta_1 \), it is a transient state, whereas when \( 0 < |s_a| < \zeta_1 \), it is a steady state. The output amount of the improved compensation controller at the transient state is much larger than that at the steady state, which can effectively ameliorate the shortcoming of poor response of the self-organizing recurrent cerebellar model articulation controller at the transient state. From Equation (77), we see that if the selected \( \delta_i \) still meets \( |\gamma_1| < \delta_i \) and an appropriate \( \gamma_1 \) is selected, the stability of the overall system will not change. Introducing ATFSORC with an improved compensation controller still can guarantee the convergence in errors of the system and maintain the stability of the system.

3.5. Design of Adaptive TSK Fuzzy Self-Organizing Recurrent Cerebellar Model Articulation Controller for Chaotic Stability Control System

In this study, the proposed adaptive TSK fuzzy self-organizing recurrent cerebellar model articulation controller is used as a stability controller for chaotic systems. Therefore, by converting the controlled body in Figure 9 into a chaotic system, \( y_t \) can be regarded as the chaotic system outputs of \( x(k) \) and \( y(k) \); \( y \) can be regarded as the chaotic system delayed
feedback outputs of \( x(k-1) \) and \( y(k-1) \); \( e_x \) and \( e_y \) are errors of \( x(k) - x(k-1) \) and \( y(k) - y(k-1) \), respectively. Moreover, in order to reduce the architectural complexity of ATFSORC, the two errors \( e_x \) and \( e_y \) are merged into one error \( v \) through the signed distance.

\[ v(t) = \frac{(\lambda_2 x(t) + e_x(t))/\sqrt{1 + \lambda_2^2} + (\lambda_2 y(t) - e_y(t))/\sqrt{1 + \lambda_2^2}} {\lambda_2} \]  

Equation (82)

Differentiating Equation (82) and substituting Equation (81) into it, we get Equation (83).

\[ v(t) = \frac{(\lambda_2 x(t) + e_x(t))/\sqrt{1 + \lambda_2^2} + (\lambda_2 y(t) - e_y(t))/\sqrt{1 + \lambda_2^2}} {\lambda_2} \]

Equation (83)

**Figure 9.** Architectural diagram of ATFSORC stability controller for chaotic system.

First, the chaotic system is expressed as Equation (78)

\[
\begin{align*}
\dot{x}(t) &= x(t)(x(t) - 1)(1 - rx(t)) - y(t) + I(t) + f(t) \\
\dot{y}(t) &= bx(t)
\end{align*}
\]

Equation (78)

After Equation (78) goes through the time delay, Equation (79) can be obtained.

\[
\begin{align*}
\dot{x}(t - \tau) &= x(t - \tau)(x(t - \tau) - 1)(1 - rx(t - \tau)) - y(t - \tau) + I(t - \tau) + f(t - \tau) \\
\dot{y}(t - \tau) &= bx(t - \tau)
\end{align*}
\]

Equation (79)

The errors are defined as

\[
\begin{align*}
e_x(t) &= x(t) - x(t - \tau) \\
e_y(t) &= y(t) - y(t - \tau)
\end{align*}
\]

Equation (80)

Next, the error dynamic system can be expressed as Equation (81).

\[
\begin{align*}
\dot{e}_x(t) &= x(t) - \dot{x}(t - \tau) \\
&= x(t)(x(t) - 1)(1 - rx(t)) - y(t) + I(t) + f(t) \\
&\quad - [x(t - \tau)(x(t - \tau) - 1)(1 - rx(t - \tau))] \\
&\quad - y(t - \tau) + I(t - \tau) + f(t - \tau) \\
\dot{e}_y(t) &= y(t) - \dot{y}(t - \tau) \\
&= bx(t) - x(t - \tau)
\end{align*}
\]

Equation (81)

Merging the two errors of Equation (80) through the signed distance into an error \( v \), we get Equation (82).

\[
\begin{align*}
v(t) &= (\lambda_2 x(t) + e_x(t))/\sqrt{1 + \lambda_2^2} \\
&\quad + (\lambda_2 y(t) - e_y(t))/\sqrt{1 + \lambda_2^2}
\end{align*}
\]

Equation (82)

Differentiating Equation (82) and substituting Equation (81) into it, we get Equation (83).

\[
\begin{align*}
v(t) &= (\lambda_2 x(t) + e_x(t))/\sqrt{1 + \lambda_2^2} \\
&\quad + (\lambda_2 y(t) - e_y(t))/\sqrt{1 + \lambda_2^2} \\
&\quad + \lambda_2 x(t)(x(t) - 1)(1 - rx(t)) - y(t) + I(t) + f(t - \tau)(x(t - \tau) - 1)(1 - rx(t - \tau)) \\
&\quad + y(t - \tau) - I(t - \tau) - f(t - \tau) + b(x(t) - x(t - \tau))/\sqrt{1 + \lambda_2^2} \\
&\quad - \lambda_2 y(t) + (f(t) - f(t - \tau))/\sqrt{1 + \lambda_2^2} + \lambda_2 x(t)(x(t) - 1)(1 - rx(t)) + I(t) \\
&\quad - x(t - \tau)(x(t - \tau) - 1)(1 - rx(t - \tau)) - I(t - \tau) + b(x(t) - x(t - \tau)) + \lambda_2 x(t) \\
&\quad - \lambda_2 y(t - \tau))/\sqrt{1 + \lambda_2^2} \\
&\quad = Av(t) + B(f(t) - f(t - \tau)) + d
\end{align*}
\]

Equation (83)
where
\[ A = -\lambda_2, \]
\[ B = \sqrt{1 + \lambda_2^2}, \]
\[ d = [\lambda_2 \{ x(t)(x(t) - 1)(1 - rx(t)) + I(t) - x(t - \tau)x(t - \tau - 1) \}
\[ \quad + (1 - rx(t - \tau)) + b(x(t) - x(t - \tau)) + \lambda_2^2 x(t) - \lambda_2^2 x(t - \tau)]/\sqrt{1 + \lambda_2^2}. \]

In an actual chaotic system, considering the dynamics of the chaotic system, the parameters will change accordingly under different conditions. Therefore, Equation (88) can be rewritten as
\[ \dot{v}(t) = (A_\nu v(t) + (B_\nu + \Delta B)(f(t) - f(t - \tau))) + (d + \Delta d), \]
\[ = A_\nu v(t) + B_\nu f(t) - f(t - \tau) + \Delta A v(t) + \Delta B f(t) - f(t - \tau)) + d_\nu, \]
\[ = A_\nu v(t) + B_\nu f(t) - f(t - \tau) + d_\nu. \]
where \( A_\nu \) and \( B_\nu \) are the nominal values of \( A \) and \( B \), respectively; \( \Delta A \) and \( \Delta B \) are the variation values of \( A \) and \( B \), respectively; and \( d_\nu \) and \( d_\nu \) are the uncertainty dynamic model (system uncertainty), where \( d_\nu = d + \Delta d \) and \( d_\nu = \Delta A v + \Delta B f(t) - f(t - \tau) + d_\nu \). By transposing and arranging Equation (84), we get
\[ f(t) - f(t - \tau) = \frac{1}{B_\nu}[\dot{A}_\nu v(t) - d_\nu + \dot{v}(t)]. \]

From Equation (85), the proposed ATFSORC output control amount \( u \) can be assigned as
\[ u = f(t) - f(t - \tau); \]
rearranging this can give
\[ u = \frac{1}{B_\nu}[\dot{A}_\nu v(t) - d_\nu + \dot{v}(t)]. \]

Ideally, if all the parameters of the chaotic system are known, an ideal control quantity \( u^* \) can be constructed as
\[ u^* = \frac{1}{B_\nu}[\dot{A}_\nu v(t) - d_\nu + \dot{v}^*(t) - k_\nu e(t)], \]
where \( v^* \) is the ideal error amount and \( k_\nu \) is a normal number. Error \( e \) is defined as Equation (88).
\[ e = v - v^*. \]

If the control amount \( u \) approaches the ideal control amount \( u^* \), Equation (89) will be obtained based on Equations (86) and (87), and Equation (90) will be obtained from Equations (88) and (89).
\[ \dot{v} = v^* - k_\nu e, \]
\[ \dot{e} + k_\nu e = 0. \]
Since \( k_\nu \) is a positive real number, and Equation (90) is a strictly Hurwitz polynomial, the error will converge to zero as time passes to infinity, i.e., \( \lim_{t \to \infty} e = 0 \). Next, the integrated error function \( s_b \) is defined as in Equation (91):
\[ s_b(t) = e(t) + k_\nu \int_0^t e(\tau) d\tau. \]

Differentiating Equation (91) with respect to time \( t \), we get
\[ \dot{s}_b = \dot{e} + k_\nu e. \]

When the control amount is an ideal control amount \( u^* \), error \( e \) can become asymptotically stable. However, in fact, such an ideal control amount cannot be obtained, because system parameters and the unknown dynamic model will vary along with many uncertain factors, and these cannot be known precisely in advance. Therefore, this study proposes ATFSORC that can use the good approximation capability of TFSORC to learn the ideal
control amount \( u^* \), and use the compensation controller to compensate for the error between TFSORC and \( u^* \). Figure 9 is the ATFSORC architectural diagram for chaotic system synchronization; its output is

\[
\mathbf{u} = \mathbf{u}_\text{TFSORC} + \mathbf{u}_c ,
\]

(93)

where \( \mathbf{u} \) is the total control amount; \( \mathbf{u}_\text{TFSORC} \) is the output amount of TFSORC; and \( \mathbf{u}_c \) is the output amount of the compensation controller.

Next, the association memory space is partitioned so that the optimal number of stage layers is \( p^* \), which is divided into an activated part as \( p \) and an inactivated part as \( p^* - p \). Therefore, the optimal parameters \( \Lambda^*, W^*, m^*, v^*, \) and \( r^* \) are also divided into two parts.

\[
\Lambda^* = \left[ \begin{array}{c} \Lambda_a^* \\ \Lambda_i^* \end{array} \right],
\]

(94)

\[
W^* = \left[ \begin{array}{c} W_a^* \\ W_i^* \end{array} \right],
\]

(95)

\[
m^* = \left[ \begin{array}{c} m_a^* \\ m_i^* \end{array} \right],
\]

(96)

\[
v^* = \left[ \begin{array}{c} v_a^* \\ v_i^* \end{array} \right],
\]

(97)

\[
r^* = \left[ \begin{array}{c} r_a^* \\ r_i^* \end{array} \right],
\]

(98)

where \( \Lambda^*, W^*, m^*, v^*, \) and \( r^* \) are the optimal parameters corresponding to \( \Lambda, W, m, v, \) and \( r \), respectively; \( \Lambda_a, W_a, m_a, v_a, \) and \( r_a \) represent the activated parts, whereas \( \Lambda_i, W_i, m_i, v_i, \) and \( r_i \) represent the inactivated parts.

Based on the universal approximation theorem, an optimal output control amount for a TSK fuzzy self-organizing recurrent cerebellar model articulation controller, \( \mathbf{u}_\text{TFSORC}^* \), is assumed, and the ideal control amount \( u^* \) is expressed as Equation (99).

\[
u^* = \mathbf{u}_\text{TFSORC}^*(x, W^*, m^*, v^*, r^*) + \rho = W^* \Lambda^* + \rho ,
\]

(99)

where \( \rho \) represents the error between the two control amounts. Since the optimal output control amount \( u^*_\text{TFSORC} \) cannot be obtained easily, \( u^*_\text{TFSORC} \) is estimated in an iterative manner. That is, we assign

\[
u = \mathbf{u}_\text{TFSORC}(x, \hat{W}_a, \hat{m}_a, \hat{v}_a, \hat{r}_a) + \mathbf{u}_c \equiv \hat{W}_a \hat{\Lambda}_a + \mathbf{u}_c ,
\]

(100)

where \( \hat{\Lambda}, \hat{W}, \hat{m}, \hat{v}, \hat{r} \) are estimated values corresponding to \( \Lambda, W, m, v, \) and \( r \), respectively. Next, subtracting Equation (86) from Equation (87), we get

\[
u^* - u = \frac{1}{B_a}(\hat{\nu}^* - \hat{\nu} - k_e e), \]

\[
s_e = \hat{e} + k_e e = B_ac[u^* - u].
\]

(101)

Substituting Equations (99) and (100) into the formula, Equation (101) can be rewritten as...
\[
\dot{s}_a = B_s[u^* - u] \\
= B_s(W_s^T \dot{A}_s + W_s^T \dot{A}_s^* + \rho - \dot{W}_s^T \dot{\hat{A}}_s + u_s) \\
= B_s(W_s^T \dot{A}_s - \hat{W}_s^T \dot{A}_s + W_s^T \dot{A}_s^* + \rho - u_s) \\
= B_s[(W_s^T - \hat{W}_s^T) \dot{A}_s^* + \hat{W}_s^T (\dot{A}_s^* - \hat{A}_s) + W_s^T \dot{A}_s^* + \rho - u_s] \\
= B_s[\hat{W}_s^T \dot{A}_s^* + \hat{W}_s^T \dot{\hat{A}}_s + W_s^T \dot{A}_s^* + \rho - u_s]
\]

(102)

where \( \tilde{A}_a = A_s^* - \hat{A}_a \); \( \tilde{W}_a = W_a^* - \hat{W}_a \)

Through the Taylor series, the following expansions can be obtained.

\[
\tilde{\Lambda}_a = \Lambda_s^* - \hat{\Lambda}_a = \begin{bmatrix}
\lambda_1^* - \hat{\lambda}_1 \\
\lambda_2^* - \hat{\lambda}_2 \\
\vdots \\
\lambda_p^* - \hat{\lambda}_p
\end{bmatrix} = \begin{bmatrix}
\frac{\partial b_{1}^{T} f}{\partial \alpha} \\
\frac{\partial b_{2}^{T} f}{\partial \alpha} \\
\vdots \\
\frac{\partial b_{p}^{T} f}{\partial \alpha}
\end{bmatrix}\begin{bmatrix}
\Delta m_{a} \\
\Delta m_{a}^* \\
\vdots \\
\Delta m_{a}
\end{bmatrix} + \begin{bmatrix}
\frac{\partial b_{1}^{T} f}{\partial \alpha} \\
\frac{\partial b_{2}^{T} f}{\partial \alpha} \\
\vdots \\
\frac{\partial b_{p}^{T} f}{\partial \alpha}
\end{bmatrix}\begin{bmatrix}
\Delta v_{a} \\
\Delta v_{a}^* \\
\vdots \\
\Delta v_{a}
\end{bmatrix} (v_{a}^* - \hat{v}_{a})
\]

(103)

where \( \mathbf{O} \) is a vector of higher-order components. For convenience of derivation, assigning

\[
\tilde{m}_a = m_a^* - \hat{m}_a, \quad \tilde{v}_a = v_a^* - \hat{v}_a, \quad \tilde{r}_a = r_a^* - \hat{r}_a
\]

and

\[
\mathbf{C} = \begin{bmatrix}
\frac{\partial b_{1}^{T} f}{\partial \alpha}, (\frac{\partial b_{2}^{T} f}{\partial \alpha}), \ldots, (\frac{\partial b_{p}^{T} f}{\partial \alpha})
\end{bmatrix} \\
\mathbf{G} = \begin{bmatrix}
\frac{\partial b_{1}^{T} f}{\partial \alpha}, (\frac{\partial b_{2}^{T} f}{\partial \alpha}), \ldots, (\frac{\partial b_{p}^{T} f}{\partial \alpha})
\end{bmatrix} \\
\mathbf{H} = \begin{bmatrix}
\frac{\partial b_{1}^{T} f}{\partial \alpha}, (\frac{\partial b_{2}^{T} f}{\partial \alpha}), \ldots, (\frac{\partial b_{p}^{T} f}{\partial \alpha})
\end{bmatrix}
\]

(104) (105) (106)

then, Equation (103) can be rewritten as

\[
\tilde{\Lambda}_a = \mathbf{C}^{T} \tilde{m}_a + \mathbf{G}^{T} \tilde{v}_a + \mathbf{H}^{T} \tilde{r}_a + \mathbf{O}_f.
\]

(107)

Since \( \tilde{\Lambda}_a = \Lambda_a^* - \hat{\Lambda}_a \), Equation (107) can also be expressed as

\[
\Lambda_a^* = \tilde{\Lambda}_a + \mathbf{C}^{T} \tilde{m}_a + \mathbf{G}^{T} \tilde{v}_a + \mathbf{H}^{T} \tilde{r}_a + \mathbf{O}_f.
\]

(108)

Then, substituting Equations (107) and (108) into Equation (102), we get
\[
\dot{s}(t) = B_s[\dot{W}_s^T \dot{A}_s + \dot{W}_s^T \dot{\hat{A}}_s + W_s^T \dot{A}_s + \rho - u_s] \\
= B_s[D_s + \dot{W}_s^T (C^T \dot{m}_s + G^T \dot{v}_s + H^T \dot{r}_e + \dot{O}_s) \\
+ \dot{W}_s^T (C^T \dot{\hat{m}}_s + G^T \dot{\hat{v}}_s + H^T \dot{\hat{r}}_e + \dot{O}_s) + W_s^T \dot{A}_s + \rho - u_s] \\
= B_s[D_s + \dot{W}_s^T (C^T \dot{m}_s + G^T \dot{v}_s + H^T \dot{r}_e + \dot{O}_s)] \\
+ (\dot{W}_s^T C^T \dot{\hat{m}}_s + \dot{W}_s^T G^T \dot{\hat{v}}_s + \dot{W}_s^T H^T \dot{\hat{r}}_e + \dot{W}_s^T \dot{\hat{O}}_s + W_s^T \dot{A}_s + \rho - u_s] \\
= B_s[D_s + \dot{W}_s^T (B_s \dot{\hat{W}}_s + \dot{\hat{W}}_s^T C^T \dot{m}_s + G^T \dot{v}_s + H^T \dot{r}_e + \dot{O}_s) + D_s - u_s] \\
\]
where \( D_s = \dot{W}_s^T C^T \dot{m}_s + \dot{W}_s^T G^T \dot{v}_s + \dot{W}_s^T H^T \dot{r}_e + \dot{W}_s^T \dot{\hat{O}}_s + W_s^T \dot{A}_s + \rho \). Assuming that the upper limit of \( D_s \) is a normal number \( \delta_2 \)
\[
|D_s| < \delta_2. 
\]

**Theorem 2.** Considering a chaotic stability control system as Equation (84), if the stability controller is designed to be Equation (100) and the ATFSORC’s adaptive laws and compensation controller output are as shown from Equation (111) to Equation (115), then the stability of such a controlled system can be guaranteed.

\[
\dot{\hat{W}}_s = s_2 \beta_2 \dot{\hat{A}}_s, 
\]
\[
\dot{\hat{m}}_s = s_2 \beta_2 \dot{C} \dot{\hat{W}}_s, 
\]
\[
\dot{\hat{v}}_s = s_2 \beta_2 \dot{G} \dot{\hat{W}}_s, 
\]
\[
\dot{\hat{r}}_s = s_2 \beta_2 \dot{H} \dot{\hat{W}}_s, 
\]
\[
u_c = \delta_2 \text{sgn}[\dot{s}_s], 
\]

**Proof of Theorem 2.** Using the Lyapunov theorem to derive the adaptive laws, the Lyapunov function candidate \( V_c \) is first defined as follows:

\[
V_c = \frac{1}{2} \dot{s}_s^T s_2 \beta_2 \dot{A}_s + \frac{1}{2} \dot{\hat{m}}_s^T s_2 \beta_2 \dot{\hat{A}}_s + \frac{1}{2} \dot{\hat{v}}_s^T s_2 \beta_2 \dot{\hat{V}}_s + \frac{1}{2} \dot{\hat{r}}_s^T s_2 \beta_2 \dot{\hat{R}}_s 
\]

where \( \beta_2 \) is a normal number greater than zero. By differentiating Equation (116) and substituting Equation (109) into the equation, we get

\[
\dot{V}_c = s_2 \dot{\hat{W}}_s^T \dot{A}_s - B_s \dot{W}_s^T \dot{A}_s + \frac{1}{2} \dot{\hat{m}}_s^T \dot{\hat{m}}_s + \frac{1}{2} \dot{\hat{v}}_s^T \dot{\hat{v}}_s + \frac{1}{2} \dot{\hat{r}}_s^T \dot{\hat{r}}_s 
\]

As \( V_c \geq 0 \) and in order to achieve the goal of \( \dot{V}_c < 0 \), we can assign

\[
\dot{W}_s^T \left[ s_2 \beta_2 \dot{A}_s - B_s \dot{\hat{W}}_s \right] = 0, 
\]
\[
\begin{align*}
\left[ s_t B_a \tilde{W}_a^T \tilde{C}^T - \frac{B_a \tilde{m}_a}{\beta_{s_a}} \right] \tilde{m}_a &= 0, \\
\left[ s_t B_a \tilde{W}_a^T \tilde{G}^T - \frac{B_a \tilde{v}_a}{\beta_{s_a}} \right] \tilde{V}_a &= 0, \\
\left[ s_t B_a \tilde{W}_a^T \tilde{H}^T - \frac{B_a \tilde{\dot{v}}_a}{\beta_{s_a}} \right] \tilde{P}_a &= 0.
\end{align*}
\]

Substituting all formulas from Equation (111) to Equation (115) into Equation (117), we get
\[
\dot{V}_{s_a} = s_t B_a (D_2 - \delta_s \text{sgn}(s_a)) \\
= s_t B_a D_2 - |s_a| \beta_{s_a} \delta_s \leq -(\delta_s - [D_2] \beta_{s_a} |s_a| \leq 0.
\]

From Equation (71), as \( \dot{V}_{s_a} \leq 0 \) is a negative semi-definite function, there is no guarantee that the error will converge to zero. Using Barbalat’s Lemma, we can prove that when time approaches infinity, \( \dot{V}_{s_a}(t) \) approaches zero.

The function \( \Xi(t) \) is defined as follows:
\[
\Xi(t) = (\delta_s - [D_2] \beta_{s_a} |s_a| \leq -\dot{V}_{s_a}(t).
\]

Integrating Equation (123) with respect to time, we get Equation (124), where \( V_{s_a}(t) \) is as shown in Equation (116).
\[
\int_{t_0}^{t} \Xi(t) \, d\tau \leq V_{s_a}(0) - V_{s_a}(t)
\]

Since \( V_{s_a}(0) \) is bounded, while \( V_{s_a}(t) \) is bounded and a non-incremental function, it can be determined from Equation (124) that \( \Xi(t) \) is bounded by time, i.e.,
\[
\lim_{t \to +\infty} \int_{t_0}^{t} \Xi(t) \, d\tau < \infty.
\]

Next, since \( V_{s_a}(t) > 0 \) and \( \dot{V}_{s_a}(t) \leq 0 \), Equation (116) shows that \( \tilde{W}_a, \tilde{m}_a, \tilde{v}_a, \text{ and } \tilde{r}_a \) are bounded. As inferred from Equation (122) that when \( s_a \neq 0 \), from Equation (109), we see that \( \tilde{s}_a \) is bounded and \( s_a \) is uniformly continuous; Equation (122) shows that \( \tilde{V}_a(t) \) is bounded, thus proving that \( \dot{V}_{s_a}(t) \) is uniformly continuous and \( \dot{V}_{s_a}(t) \) is bounded. From Barbalat’s Lemma, we can further obtain the following.
\[
\lim_{t \to +\infty} \dot{V}_{s_a}(t) = 0, \forall s_a \neq 0.
\]

From Equation (122), we see that when \( s_a = 0 \), \( \dot{V}_{s_a}(t) = 0 \). From the above-mentioned inference, it can be learnt that for arbitrary \( s_a \) when the time \( t \) approaches infinity, \( \dot{V}_{s_a}(t) \) will converge to zero, and \( s_a \) at this point of time will also converge to zero. Equation (92) shows that if an appropriate \( k_0 \) is selected to conform to the Hurwitz polynomial, the error \( e \) and \( v \) will also converge to zero when the time approaches infinity. Through the above derivations, we can ensure that the designed adaptive TSK fuzzy self-organizing recurrent cerebellar model articulation controller system is stable. □

3.6. Design of Improved Compensator

The adaptive TSK fuzzy self-organizing cerebellar model articulation controller proposed in this study contains an improved compensation controller. From Equation (115), the output of compensation controller \( u_0 \) is composed of a normal number \( \delta_e \) and a sign function of \( s_0 \) wherein the function of \( \text{sgn}(s_0) \) is defined as in Equation (127). Since \( \delta_e \) is a normal number, the output of the compensation controller depends on the integrated error function \( s_0 \) value. In fact, there are only three types of states as \( \delta_e, 0, \) and \(-\delta_e \), which
cause the shortcoming of insufficient transient-state compensation or excessive steady-state compensation.

\[
\text{sgn}[s_b] = \begin{cases} 
1 & , s_b > 0 \\
0 & , s_b = 0 \\
-1 & , s_b < 0
\end{cases}
\]  \hspace{1cm} (127)

Therefore, this study proposes an improved compensator to ameliorate the original compensation controller as presented arithmetically in Equation (115). Design of the improved compensation controller focuses on allowing the compensation controller to provide a larger compensation amount in ATFSORC at the transient state and a smaller compensation amount after ATFSORC enters the steady state.

Errors will increase when the compensation amount provided by the compensation controller is insufficient; therefore, the integrated error function \( s_0 \) can be used as the basis to determine whether ATFSORC has completed learning and entered the steady state. The above-mentioned design concept can be implemented from Equation (128).

\[
u_c = \begin{cases} 
\delta_2 \gamma_2 s_b & , |s_b| > \zeta_2 \\
\delta_2 \text{sgn}[s_b] & , 0 < |s_b| < \zeta_2
\end{cases}
\]  \hspace{1cm} (128)

where \( \zeta_2 \) is the threshold value for switching compensation control quantity; \( \gamma_2 \) is a compensation coefficient; and \( \gamma_2 > 1 \), which is used to elastically adjust the magnitude of compensation amount at the transient state. When \( |s_b| > \zeta_2 \), it is a transient state, while when \( 0 < |s_b| < \zeta_2 \), it is a steady state. Furthermore, the output amount of the improved compensation controller at the transient state is considerably larger than that at the steady state, which can effectively ameliorate the shortcoming of poor response of the self-organizing recurrent cerebellar model articulation controller at the transient state. By observing (128), note that if the selected \( \delta_2 \) still meets \( |s| < \delta_2 \) and an appropriate \( \gamma_2 \) is selected, the stability of the overall system will not change. Introducing ATFSORC with an improved compensation controller can still guarantee the convergence in errors of the system and maintain the stability of the system.

4. Simulation Results and Analysis on Chaotic Synchronization and Stability Control

In this study, the proposed ATFSORC was applied on the synchronization control of chaotic systems, and it was compared with CMAC and fuzzy CMAC (FCMAC) to validate the control performance of the proposed ATFSORC. Relevant parameters for simulation are shown in Table 2. At the beginning of the simulation, a trial-and-error method was used to find a set of initial values of controller parameters that can enable the system to run smoothly. After that, all the controller parameters were subjected to simulation using these initial values. The simulation contains the following two scenarios:

1. By adjusting the coupling strength \( g \) of the gap junction in Equations (28) and (29) to 0.01 and 1 as well as by changing the cut-in control time point of ATFSORC to 300 s after the chaotic system running, the simulation would validate the feasibility of the controller proposed in this study.

2. By comparing the response data of ATFSORC with those of CMAC and FCMAC, the controller performance was determined via root mean square errors and average errors.
Table 2. Simulation parameters for the synchronization control of chaotic systems.

| Relevant Parameters of Simulation | Parameter Setting Values |
|----------------------------------|--------------------------|
| System sampling time             | 0.01 s;                  |
| Signed distance parameter        | $\lambda_1 = 1$;         |
| Chaotic system parameters        | $\omega = 0.7986; A = 0.1; b = 1; r = 10; f = 0.1271$; |
| Initial point of chaotic system  | $(x_1, x_2, y_1, y_2) = (1, 0, 1, 0);$ |
|                                  | $\alpha_0 = 0.98 (g = 0.01);$ |
|                                  | $\alpha_n = 0.98 (g = 1);$  |
|                                  | $\alpha_0 = 0.98 (g = 0.01, cut in at 300 s);$ |
|                                  | $\beta_0 = 0.4 (g = 0.01);$ |
|                                  | $\beta_0 = 0.4 (g = 1);$  |
|                                  | $\beta_0 = 0.35 (g = 0.01, cut in at 300 s);$ |
| Relevant parameters of ATFSORC   | $m_1 = 0; v_1 = 0.1; k_1 = 1.8;$ |
|                                  | $\beta_{at1} = 1 \times 10^{-6}; \beta_{at1} = 1 \times 10^{-3};$ |
|                                  | $\beta_{at1} = 8 \times 10^{-6}; \beta_{at1} = 8 \times 10^{-6}.$ |

Figures 10 and 11 show the uncontrolled chaotic system response diagrams for coupling strengths $g = 0.01$ and $g = 1$, respectively. To observe the unpredictable behavior of the chaotic system, the control amount $u$ for the chaotic system in Equation (29) was set to zero, and the initial condition was set to $(x_1, x_2, y_1, y_2) = (1, 0, 1, 0)$. Figures 10a and 11a show the phase diagrams of the master system. The phase diagrams show that the master system indeed operates at a chaotic state. Figures 10b and 11b show the phase diagrams of the slave system, which have different initial starting points from those of the master system; thus, the diagrams demonstrate the characteristics of chaos and have phase plane trajectories different from those of the master system. Figure 10c shows a phase diagram constructed by $x_1$ and $x_2$. As shown very clearly from the diagram, when the primary and secondary systems are asynchronous, the entire phase plane trajectory is disorganized and in disorder. Figure 11c shows a phase diagram constructed by $x_1$ and $x_2$. As shown very clearly from the diagram, when the primary and secondary systems are asynchronous, the entire phase plane is slightly disorganized; however, due to the large coupling strength, the $x$ coefficients of the master–slave system will gradually synchronize. Figure 10d shows a phase diagram constructed by $y_1$ and $y_2$. Similarly, we can clearly discover that when the control amount is zero, the asynchronous state is presented. Figure 11d shows the phase diagram constructed of $y_1$ and $y_2$. We can similarly discover that due to the large coupling strength, the $y$ coefficient for the master–slave system will gradually synchronize.

Figures 12 and 13 show the response diagrams of chaotic system synchronization for ATFSORC with coupling strengths of $g = 0.01$ and $g = 1$, respectively; initial conditions were set as $(x_1, x_2, y_1, y_2) = (1, 0, 1, 0)$. Figures 12a and 13a show the phase diagrams of the master system, starting from the coordinate at $(1, 1)$ and performing a chaotic counterclockwise trajectory motion. Figures 12b and 13b show the phase diagram of the slave system, starting from the coordinates at $(0, 0)$; it can be seen that after inserting the controller, the slave system can rapidly catch up with the trajectory of the master system. Figures 12c and 13c show the phase diagrams for $x_1$ and $x_2$. The observation shows that due to the addition of ATFSORC, the $x$ parameters for the master–slave system can be synchronized in a short period of time, demonstrating excellent control performance. Figures 12d and 13d show the phase diagrams for $y_1$ and $y_2$; we can observe that the slave system can be rapidly synchronized with the master system. Figures 12e and 12f show the response diagram and errors for $x_1$ and $x_2$, respectively. The observation implies that the master–slave system can be fully synchronized in about 9 s, and the error after the steady
state is almost approaching zero. Figure 13e,f are the response diagram and errors for $x_1$ and $x_2$, respectively. The observation shows that the master–slave system can be fully synchronized in about 9 s, and the error after the steady state is almost approaching zero. Figure 12g,h show the response diagram and errors for $y_1$ and $y_2$. In the same manner, it can be observed that the synchronization can be complete in about 9 s, and the steady-state error approaches zero. Looking coordinately with the number of associate memory layers in Figure 12i,j, it is found that because the transient error is larger, the number of layers is increased from the preset one layer to six layers only within 0.3 s, the number of layers is increased layer by layer, and the number of association memory layers at the steady state is fixed as 7 layers. Figure 13g,h show the response diagram and errors for $y_1$ and $y_2$. In the same manner, it can be observed that the synchronization can be completed in about 9 s, and the steady-state error approaches zero. Looking coordinately with the number of associate memory layers in Figure 13i,j, it can be found that because the transient error is larger, the number of layers increases from the preset one layer to six layers only within 0.2 s, and the number of associate memory layers at steady state is fixed as seven layers. Figures 12k and 13k show the response diagrams of all control amounts of ATFSORC, wherein adding the improved compensation controller output $u_i$ on the TSK fuzzy self-organizing recurrent cerebellar model articulation controller output $u_{TFSORC}$ will be equal to the control amount $u$ in Equation (29).

Figure 14 shows the response diagrams of the chaotic system for ATFSORC, wherein the coupling strength $g = 0.01$ and $g = 1$; the chaotic system is first allowed to run for 300 s before the controller cuts in, and the initial conditions are set as $(x_1, x_2, y_1, y_2) = (1, 0, 1, 0)$. Figure 14a,b show the $x_1$, $x_2$ response and $y_1$, $y_2$ response for ATFSORC, respectively. We can see that the master–slave system is at an asynchronous stage before 300 s; after approximately 6 s, the controller cuts in, and the slave system can keep up with the master system trajectory. Figure 14c,d show errors $e_1$ and $e_2$ respectively. Looking coordinately with the number of association memory layers and their partial enlarged views in Figure 14e,f, it is found that because the errors at the transient state are larger, the number of layers is added from the preset one layer to six layers only within 0.25 s; after 0.5 s, the error is smaller, the number of layers is reduced by one layer. Figure 14g shows the response diagram of all the control amount for ATFSORC, wherein adding the improved compensation controller output $u_i$ on the TSK fuzzy self-organizing recurrent cerebellar model articulation controller output $u_{TFSORC}$ will be equal to the control amount $u$ in Equation (29).

Figure 15 compares the response diagrams of chaotic system synchronization for ATFSORC, FCMAC, and CMAC, with a coupling strength of $g = 0.01$ and an initial condition set as $(x_1, x_2, y_1, y_2) = (1, 0, 1, 0)$. Figure 15a–c show response diagrams for the master systems; Figure 15d–f show comparisons of response diagrams for the slave systems. The comparisons show that although all the slave systems can catch up with the trajectories of the master systems, the trajectory jitters of CMAC and FCMAC at the transient state are more obvious than that of ATFSORC. Figure 15g–i show comparisons of the $x_1$ and $x_2$ phase diagrams. Under the ideal circumstance, when the steady-state error is 0, i.e., $x_1$ is equal to $x_2$, the phase diagram should present a strip of a straight line; the line strip from CMAC is thicker, and its transient time is longer; FCMAC is second, and ATFSORC is the best. Figure 15j–l show comparisons of the $y_1$ and $y_2$ phase diagrams. Under the ideal circumstance, when the steady-state error is 0, i.e., $y_1$ is equal to $y_2$, the phase diagram should present a strip of a straight line; the comparisons show that the line strip of CMAC is thicker and the transient time is longer, FCMAC is second, and ATFSORC is the best.

Figure 16 compares the response diagrams of chaotic system synchronization for AFSSORC, FCMAC, and CMAC, with a coupling strength of $g = 1$ and the initial condition set as $(x_1, x_2, y_1, y_2) = (1, 0, 1, 0)$. Figure 16a–c show comparisons of the master systems’ response diagrams. Figure 16d–f show comparisons of the slave systems’ response
The comparisons indicate that although all the slave systems can catch up with the trajectories of the master systems, the trajectory jittering phenomena of CMAC and FCMAC at the transient state are more obvious than that of ATFSORC. Figure 16g–i show comparisons of the $x_1$ and $x_2$ phase diagrams. Under the ideal circumstance, when the steady-state error is $0$, i.e., $x_1$ is equal to $x_2$, the phase diagram should present a strip of a straight line. The comparisons show that the CMAC line strip is thicker and the transient time is longer, FCMAC is second, and ATFSORC is the best. Figure 16j–l show comparisons of $y_1$ and $y_2$ phase diagrams. Under the ideal circumstance, when the steady-state error is $0$, i.e., $y_1$ is equal to $y_2$, the phase diagram should present a strip of a straight line. The comparison shows that the transient time for CMAC is longer, FCMAC is second, and ATFSORC is the best. Figure 16m–o show comparisons of the $x_1$ and $x_2$ response diagram. Coordinating with the comparisons of error $e_x$ from Figure 16p–r, we can see that although all the slave systems can catch up with the trajectories of the master systems, the steady-state error of CMAC lies between $-0.002$ and $0.002$, and the steady-state error of FCMAC lies between $-0.0002$ and $0.0002$, whereas the steady-state error of ATFSORC is almost $0$. Therefore, the proposed ATFSORC error is smaller and the convergence velocity is faster.

Figure 17 compares the response diagrams of chaotic system synchronization for ATFSORC, FCMAC, and CMAC, with a coupling strength of $g = 0.01$ and $g = 1$. Moreover, the chaotic system is first allowed to run for $300$ s before the controller cuts in, and the initial conditions are set as $(x_1, x_2, y_1, y_2) = (1, 0, 1, 0)$. Figure 17a–c show comparisons of $x_1$ and $x_2$ response diagrams. To demonstrate the characteristics of chaos, the controller is first removed after the system is run for $300$ s; then, the controller cuts in, thereby validating the performance of the controller. The results show that all the slave systems’ trajectories can keep up with the master systems’ trajectories, but the steady-state response of ATFSORC is better than those of CMAC and FCMAC. Figure 17d–f show comparisons of error $e_x$. The steady-state errors lie between $-0.002$ and $0.002$ for CMAC and between $-0.0002$ and $0.0002$ for FCMAC, whereas the steady-state error of ATFSORC is almost $0$; therefore, the proposed ATFSORC performance is better. Figure 17g–i show a comparison of $y_1$ and $y_2$ response diagrams. To demonstrate the characteristics of chaos, the controller is first removed after the system is run for $300$ s, and then the controller cuts in, thereby validating the performance of the controller. The results show that all the slave systems’ trajectories can keep up with the master systems’ trajectories, but the steady-state response of ATFSORC is better than those of CMAC and FCMAC. Figure 17j–l show comparisons of the error $e_y$. The steady-state error of CMAC lies between $-0.002$ and $0.002$, and that of FCMAC lies between $-0.0002$ and $0.0002$; in contrast, the steady-state error of ATFSORC is almost $0$. Therefore, the proposed ATFSORC performance is better.
Figure 10. Uncontrolled coupled nonlinear chaotic system ($\gamma = 0.01$): (a) Master system phase diagram; (b) Slave system phase diagram; (c) $x_1, x_2$ phase diagram; (d) $y_1, y_2$ phase diagram.

Figure 11. Uncontrolled coupled nonlinear chaotic system ($\gamma = 1$): (a) Master system phase diagram; (b) Slave system phase diagram; (c) $x_1, x_2$ phase diagram; (d) $y_1, y_2$ phase diagram.
Figure 12. Simulation results of ATFSORC chaotic system synchronization control (g = 0.01): (a) Master system phase diagram for ATFSORC; (b) Slave system phase diagram for ATFSORC; (c) x₁, x₂ phase diagram for ATFSORC; (d) y₁, y₂ phase diagram for ATFSORC; (e) x₁, x₂ response for ATFSORC; (f) Error e₁ for ATFSORC; (g) y₁, y₂ response for ATFSORC; (h) Error e₂ for ATFSORC; (i) Number of association memory layers for ATFSORC; (j) Number of association memory layers for ATFSORC (partial); (k) Control amount for ATFSORC.
Figure 13. Simulation results of ATFSORC chaotic system synchronization control ($g = 1$): (a) Master system phase diagram for ATFSORC; (b) Slave system phase diagram for ATFSORC; (c) $x_1$, $x_2$ phase diagram for ATFSORC; (d) $y_1$, $y_2$ phase diagram for ATFSORC; (e) $x_1$, $x_2$ response for ATFSORC; (f) Error $e_x$ for ATFSORC; (g) $y_1$, $y_2$ response for ATFSORC; (h) Error $e_y$ for ATFSORC; (i) Number of association memory layers for ATFSORC; (j) Number of association memory layers for ATFSORC (partial); (k) Control amount for ATFSORC.
Figure 14. Simulation results of ATFSORC chaotic system synchronization control ($g = 0.01$, Cut-in control at 300 sec): (a) $x_1, x_2$ response for ATFSORC; (b) error $e_1$ for ATFSORC; (c) $y_1, y_2$ response diagram for ATFSORC; (d) Error $e_1$ for ATFSORC; (e) Number of association memory layers for ATFSORC; (f) Number of association memory layers for ATFSORC (partial); (g) Control amount for ATFSORC.
Figure 15. Comparisons on simulation results of chaotic system synchronization control ($\varepsilon = 0.01$): (a) Master system phase diagram for cerebellar model articulation controller (CMAC); (b) Master system phase diagram for fuzzy CMAC (FCMAC); (c) Master system phase diagram for ATFSORC; (d) Slave system phase diagram for CMAC; (e) Slave system phase diagram for FCMAC; (f) Slave system phase diagram for ATFSORC; (g) $x_1$, $x_2$ phase diagram for CMAC; (h) $x_1$, $x_2$ phase diagram for FCMAC; (i) $x_1$, $x_2$ phase diagram for ATFSORC; (j) $y_1$, $y_2$ phase diagram for CMAC; (k) $y_1$, $y_2$ phase diagram for FCMAC; (l) $y_1$, $y_2$ phase diagram for ATFSORC.
Figure 16. Comparisons on simulation results of chaotic system synchronization control ($q = 1$): (a) Master system phase diagram for CMAC; (b) Master system phase diagram for FCMAC; (c) Master system phase diagram for ATFSORC; (d) Slave system phase diagram of CMAC; (e) Slave system phase diagram for FCMAC; (f) Slave system phase diagram for ATFSORC; (g) $x_1$, $x_2$ phase diagram for CMAC; (h) $x_1$, $x_2$ phase diagram for FCMAC; (i) $x_1$, $x_2$ phase diagram for ATFSORC; (j) $y_1$, $y_2$ phase diagram for CMAC; (k) $y_1$, $y_2$ phase diagram for FCMAC; (l) $y_1$, $y_2$ phase diagram for ATFSORC; (m) $x_1$, $x_2$ response for CMAC; (n) $x_1$, $x_2$ response for FCMAC; (o) $x_1$, $x_2$ response for ATFSORC; (p) Error $e$ for CMAC; (q) Error $e$ for FCMAC; (r) Error $e$ for ATFSORC.
Figure 17. Comparisons on simulation results of chaotic system synchronization control ($q = 0.01$, cut-in control at 300 s); (a) $x_1, x_2$ response for CMAC; (b) $x_1, x_2$ response for FCMAC; (e) $x_1, x_2$ response for ATFSORC; (d) Error $e_c$ for CMAC; (e) Error $e_c$ for FCMAC; (f) Error $e_c$ for ATFSORC; (g) $y_1, y_2$ response for CMAC; (h) $y_1, y_2$ response for FCMAC; (i) $y_1, y_2$ response for ATFSORC; (j) Error $e_y$ for CMAC; (k) Error $e_y$ for FCMAC; (l) Error $e_y$ for ATFSORC.

The proposed ATFSORC was applied on the stability control of a chaotic system, and it was compared with CMAC and FCMAC to validate the control performance of the proposed ATFSORC. Relevant parameters for simulations are shown in Table 3. At the beginning of the simulation, a set of initial values of controller parameters that allow the system to run smoothly was found by trial and error. These initial values were taken for all the controller parameters for simulation. Simulations contain the following two scenarios:

1. In the simulation, the controller will first be removed; control will cut in after the chaotic system has run for 400 s; thereby, the feasibility of the controller is validated.
2. Response data of ATFSORC will be controlled with those of CMAC and FCMAC to observe the stabilization time.

Table 3. Parameters of chaotic system stability simulation.

| Relevant Parameters of Simulation | Parameter Setting Values |
|----------------------------------|--------------------------|
| System sampling time             | 0.01 s;                  |
| Signed distance parameter        | $\lambda_2 = 1$;         |
| Chaotic system parameters        | $\omega = 0.7986$; $A = 0.1$; $b = 1$; $r = 10$; $f = 0.1271$; |
| Initial point of chaotic system  | $(x, y) = (0, 0)$; $\alpha_m = 0.98$; $\beta_m = 0.5$; |
| Relevant parameters of ATFSORC   | $m_l = 0$; $v_l = 0.1$; $k = 1.8$; $\beta_x = 1 \times 10^{-5}$; $\beta_y = 1 \times 10^{-5}$; $\beta_m = 8 \times 10^{-4}$; $\beta_m = 8 \times 10^{-4}$; |

Figure 18 shows the stability simulation results of a chaotic system for ATFSORC. In Figure 18a, to observe the characteristics of a chaotic system, the control amount of the chaotic system is set as zero, and the initial condition is set as $(x, y) = (0, 0)$. Figure 18b shows the stable phase diagram for the chaotic system. The controller cuts in at 400 s, and the trajectory of the last 100 s is plotted. It is observed that the proposed ATFSORC can immediately control the chaotic system to a stable state of period one in a short time; the phase plane presents a periodic closed trajectory. Figure 18c,e show the $x$ response diagram for ATFSORC and the $y$ response diagram for ATFSORC, respectively. After cut-in control at 400 s, the chaotic system can be stabilized in a short time, and the trajectory presents a periodic variation. Figure 18d,f show error $e_c$ for ATFSORC and error $e_y$ for ATFSORC, respectively. The coordination of the number of association memory layers in Figure 18g indicate that due to the larger transient error when the controller cuts in at 400 s, the number of associative memory layers is increased from the preset one layer to four layers within only 4 s, and then, five layers is fixed after entering the steady state. Figure 18h is a partial enlarged view on the number of association memory layers. Figure 18i is a response diagram of all control amounts for ATFSORC.

Figure 19 shows comparisons on the response diagrams of chaotic system stability for CMAC, FCMAC, and ATFSORC, wherein the control amount at the beginning is set...
as zero with cut-in control after 400 s. Figure 19a–c show the phase diagrams of uncontrolled chaotic systems. The aim is to display the characteristics of a trajectory repeat of the chaotic system. The coordination with the stability phase diagrams from Figure 19d–f shows a clear comparison of the difference before and after control. Figure 19g–i show the x response diagrams of CMAC, FCMAC, and ATFSORC. Figure 19g shows that CMAC cuts in control at 400 s and the x response should stabilize the periodic variation at about 500 s. Figure 19h indicates that FCMAC cuts in control at 400 s, and the x response should be able to stably present periodic variation in a very short time. From Figure 19i, we can see that the ATFCMAC cuts in control at 400 s, and the x response should be able to stably present the closed trajectory of period one almost immediately. Figure 19j–l show the error εt for CMAC, FCMAC, and ATFSORC. Figure 19m–o show the y response diagrams for CMAC, FCMAC, and ATFSORC. Figure 19m shows that CMAC cuts in control at 400 s, and y response should stabilize the periodic variation at about 500 s. Figure 19n shows that FCMAC cuts in control at 400 s, and the y response should be able to stably present periodic changes in a very short time. From Figure 19o, we can see that the ATFCMAC cuts in control at 400 s, and the y response should be able to stably present the closed trajectory of period one almost immediately. Figure 19p–r show the error εt for CMAC, FCMAC, and ATFSORC. From the simulation results, we know that the proposed ATFSORC possess good control performance and is suitable to be applied on the control and synchronization of chaotic systems.
Figure 18. Simulation results of ATFSORC chaotic system stability control: (a) Phase diagram for uncontrolled chaotic system; (b) Stable phase diagram for ATFSORC; (c) $x$ response for ATFSORC; (d) Error $e_x$ for ATFSORC; (e) $y$ response for ATFSORC; (f) Error $e_y$ for ATFSORC; (g) Number of association memory layers for ATFSORC; (h) Number of association memory layers for ATFSORC (partial); (i) Control amount for ATFSORC.
5. Conclusions

In this study, the TSK fuzzy system and recurrent cerebellar model with a self-organizing architecture are introduced into a cerebellar model controller in combination with an integrated error function and an improved compensation controller to design an adaptive TSK fuzzy self-organizing recurrent cerebellar model controller, ATFSORC. By adopting self-organizing and recurrent artificial neural network architecture, CMAC could possess dynamic state and parameter correction capabilities, in which the number of association memory layers can be modulated in accordance with the layer decision-making mechanism, thereby achieving better control response and the robustness resisting external disturbance. Simulation results show that when the proposed ATFSORC is applied to the synchronization and stability control of chaotic system, it possesses superior control performance when the parameters of the chaotic system or the cut-in control time point change. Furthermore, the simulation results also demonstrate that the proposed ATFSORC has better control performance in comparison with the FCMAC and CMAC.

For future research studies, it is expected that the proposed ATFSORC may have better dynamic response characteristics when combining with other advanced control theories such as function links, wavelet analysis, Petri net, etc. Moreover, the synchronization and stability control of a chaotic system proposed in this study may also be applied in conjunction with other advanced control theories for daily life applications such as the chaotic synchronous control of the number of vehicles on the highway and its gateways, cardiac pacemaker control to ensure the stabilization of heart beat frequency within a certain range under various physical conditions, etc.
Author Contributions: Contribution of each author is the same. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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
1. Lorenz, E.N. Deterministic nonperiodic flow. J. Atmos. Sci. 1963, 20, 130–141.
2. Liu, J.; Sprott, J.C.; Wang, S.; Ma, Y. Simplest chaotic system with a hyperbolic sine and its applications in DCSK scheme. IET Commun. 2018, 12, 809–815.
3. Liu, Y.; Park, J.H.; Guo, B.Z.; Shu, Y. Security-enhanced electro-optic feedback phase chaotic system based on nonlinear coupling of two delayed interfering branches. IEEE Trans. Fuzzy Syst. 2018, 26, 1040–1045.
4. Wang, C.; Ji, Y.; Wang, H.; Bai, L. Further results on stabilization of chaotic systems based on fuzzy memory sampled-data control. IEEE Photon. J. 2018, 10, 655–663.
5. Hua, Z.; Zhou, B.; Zhou, Y. Sine-transform-based chaotic system with FPGA implementation. IEEE Trans. Ind. Electron. 2018, 65, 2557–2566.
6. Tao, H.T.; Wang, J.; Deng, B.; Wei, X.L.; Chen, Y.Y. Adaptive synchronization control of coupled chaotic neurons in an external electrical stimulation. Clin. Phys. B 2013, 22, 058701.
7. Hsu, C.F. Hermite-neural-network-based adaptive control for a coupled nonlinear chaotic system. Neural. Comput. Appl. 2013, 22, 421–433.
8. Lin, F.J.; Lu, K.C.; Yang, B.H. Recurrent fuzzy cerebellar model articulation neural network based power control of a single-stage three-phase grid-connected photovoltaic system during grid faults. IEEE Trans. Ind. Electron. 2017, 64, 1258–1268.
9. Che, Y.Q.; Wang, J.S.; Zhou, S.; Deng, B. Robust synchronization control of coupled chaotic neurons under external electrical stimulation. Chaos Solitons Fractals 2009, 40, 1333–1342.
10. Wu, C.W.; Hsu, C.F.; Hwang, C.K. Master–slave chaos synchronization using adaptive TSK-type CMAC neural control. J. Frankl. Inst. 2011, 348, 1847–1868.
11. Wen, C.M.; Cheng, M.Y. Development of a recurrent fuzzy CMAC with adjustable input space quantization and self-tuning learning rate for control of a dual-axis piezoelectric actuated micromotion stage. IEEE Trans. Ind. Electron. 2013, 60, 5105–5115.
12. Lin, C.M.; Li, H.Y. TSK fuzzy CMAC-based robust adaptive backstepping control for uncertain nonlinear systems. IEEE Trans. Fuzzy Syst. 2012, 20, 1147–1154.
13. Lian, R.J. Enhanced adaptive self-organizing fuzzy sliding-mode controller for active suspension systems. IEEE Trans. Ind. Electron. 2013, 60, 958–968.
14. Lin, C.M.; Li, H.Y. Self-organizing adaptive wavelet CMAC backstepping control system design for nonlinear chaotic systems. Nonlinear Anal. Real World Appl. 2013, 14, 206–223.
15. Chen, C.S. TSK-type self-organizing recurrent-neural-fuzzy control of linear microstepping motor drives. IEEE Trans. Power Electron. 2010, 25, 2253–2265.
16. Che, Y.Q.; Cui, S.G.; Wang, J.; Deng, B.; Wei, X.L. Chaos synchronization of coupled FitzHugh-Nagumo neurons via adaptive sliding mode control. In Proceedings of the Third International Conference on Measuring Technology and Mechatronics Automation, Shanghai, China, 6–7 January 2011.
17. Peng, Y.F.; Lin, C.M. Adaptive recurrent cerebellar model articulation controller for linear ultrasonic motor with optimal learning rates. Neurocomputing 2007, 70, 2626–2637.
18. Hung, Y.C.; Lin, F.J.; Hwang, J.C. Wavelet fuzzy neural network with asymmetric membership function controller for electric power steering system via improved differential evolution. IEEE Trans. Power Electron. 2015, 30, 2350–2362.
19. Lin, C.M.; Li, H.Y. A novel adaptive wavelet fuzzy cerebellar model articulation control system design for voice coil motors. IEEE Trans. Ind. Electron. 2012, 59, 2024–2033.
20. Li, S.; Zhou, M.; Yu, X. Design and implementation of terminal sliding mode control method for PMSM speed regulation system. IEEE Trans. Industr. Inform. 2013, 9, 1879–1891.
21. Lin, F.J.; Lee, S.Y.; Chou, P.H. Computed force control system using functional link radial basis function network with asymmetric membership function for piezo-flexural nanopositioning stage. IET Control Theory Appl. 2013, 7, 2128–2142.
22. Lin, F.J.; Lu, K.C.; Ke, T.H.; Yang, B.H.; Chang, Y.R. Reactive power control of three-phase grid-connected PV system during grid faults using Takagi–Sugeno–Kang probabilistic fuzzy neural network control. IEEE Trans. Ind. Electron. 2015, 62, 5516–5528.
23. Albus, J.S. New approach to manipulator control: The cerebellar model articulation controller (CMAC). J. Dyn. Sys. Meas. Control. 1975, 97, 220–227.
24. Wang, Z.Q.; Schiano, J.L.; Ginsberg, M. Hash-coding in CMAC neural networks. IEEE Int. Conf. Neural Netw. 1996, 3, 1698–1703.
25. Sahoo, S.K.; Panda, S.K.; Xu, J.X. Iterative learning-based high-performance current controller for switched reluctance motors. *IEEE Trans. Energy Convers.* **2004**, *19*, 491–497.

26. Chuang, C.H.; Lin, Y.H.; Wang, C.H.; Hsu, C.F. Synchronization of two coupled neurons using CMAC neural networks. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, Guilin, China, 10–13 July 2011; pp. 789–794.