Multilayer Interval Type-2 Fuzzy Controller Design for Hyperchaotic Synchronization

Tien-Loc Le, (Member, IEEE)
Faculty of Mechatronics and Electronics, Lac Hong University, No. 10, Huynh Van Nghe, Bien Hoa, Dong Nai, Vietnam.
Corresponding author: Tien-Loc Le (e-mail: tienloc@lhu.edu.vn). Tel.: +84-949391937

This work was supported by Lac Hong University, Bien Hoa, Dong Nai, Vietnam.

ABSTRACT This study presents a multilayer interval type-2 fuzzy controller, which applies to synchronize the hyperchaotic systems. The main contribution of this study is the design of a multilayer structure in the membership function space, which can improve the learning ability and flexibility of the type-2 fuzzy network. Using the proposed multilayer structure, the fuzzy network can reduce the number of rules while ensuring synchronization performance. Particularly, the design of 3-dimensional Gaussian membership functions (3-DGMFs) enhances the ability to against system uncertainties and external disturbances. Moreover, the structure of the proposed network can autonomously construct by using the self-evolving algorithm. The parameters of the proposed network are online updated using an adaptive law obtained by the gradient descent method. Finally, numerical simulations on the synchronization of hyperchaotic systems are conducted to validate the efficiency and performance of the proposed method.

INDEX TERMS interval type-2 fuzzy neural network, 3-D Gaussian membership functions, self-evolving algorithm, hyperchaotic systems, adaptive control.

I. INTRODUCTION

The concept of fuzzy sets, also known as type-1 fuzzy systems (T1FSs), was proposed by Zadeh in 1965 [1]. Since then, it was attracted many researchers in various fields due to its advantages such as flexibility, intuitive knowledge, and easy computation [2-7]. The type-2 fuzzy systems (T2FSs), which can be seen as an extension of T1FSs, were developed by Zadeh in 1975 [8]. In T2FSs, the membership grade is determined by a fuzzy set, while in T1FSs, the membership grade is a crisp number. Therefore, the T2FSs can better cover the system uncertainties than the T1FSs, especially in cases where it is difficult to determine the membership functions exactly [9]. In the last decades, many contributions based on type-2 fuzzy have been applied in various fields [10-16]. The interval type-2 fuzzy systems (IT2FSs) are provided in [17] to reduce the computational cost of T2FSs. Since then, by the advantages of being easy to design and simple in computation, IT2FSs have attracted the increasing attention of many researchers [18-26]. In 2017, Herman et al. proposed an interval type-2 fuzzy logic system for handling uncertainty effects in electroencephalography (EEG) patterns [19]. In 2019, Du et al. introduced an IT2FSs sampled-data $H_\infty$ controller to control nonlinear systems [21]. In 2020, Kavikumar et al. provided a robust model reference tracking control for interval type-2 fuzzy stochastic systems [24]. Compared with traditional fuzzy logic systems, the IT2FSs have advantages such as better handling the uncertainties, more design degrees of freedom, and being more general than IT1FL [27, 28].

Determining a suitable structure for the FNN is very important, and it has a significant influence on the system performance [29]. In literature, many studies used the trial-and-error approach to obtain a suitable structure [30]. Still, it takes a long time to achieve an optimal solution, and the system performance can be further improved. Recently, self-evolving and self-organizing algorithms have been proposed to construct the network structure automatically [31-42]. These algorithms can automatically generate a new membership function if the current membership functions cannot cover the inputs well. On the other hand, if a membership function has less contribution, it can be eliminated.

Recently, the synchronization of the hyperchaotic systems has received increasing attention from many authors due to its special characteristics such as nonlinear behaviors and sensitivity to the initial values [43]. Hyperchaotic synchronization is a process that two or more hyperchaotic systems (slave hyperchaotic systems) adjust their properties for following the behaviors and bifurcations of the master hyperchaotic system. Recently, some remarkable methods for hyperchaotic systems synchronization have been provided in [44-51]. In 2017, Mohammadzadeh and Ghaemi...
proposed a new self-evolving non-singleton type-2 fuzzy neural network for secure communication [46]. In 2019, Zhou et al. introduced a dynamic analysis and finite-time synchronization of hyperchaotic systems with coexisting attractors [49]. In 2021, Singh et al. provided a multi-switching synchronization of nonlinear hyperchaotic systems via backstepping control [51]. However, most of their methods are complex, and the performance can be further improved. The hyperchaotic systems are always sensitive to initial values and disturbances. Thus, the design of an adaptive synchronizer to better scope with system uncertainties and external disturbances is necessary.

Based on the above discussions, this study presents a self-evolving multilayer interval type-2 fuzzy controller (SMIT2FC) to enhance the synchronization performance of the hyperchaotic synchronization system. In our previous work [52] a type-2 fuzzy CMAC controller was developed for chaotic system synchronization. The main improvements of this study with [52] are the design of multilayer structure and the 3-DGMFs for type-2 fuzzy controller. Compared with the studies in [52-54], the proposed SMIT2FC controller has some advantages such as the network structure can be automatically constructed; strengthen the learning ability and flexibility. In the design of our proposed SMIT2FC, the membership functions are divided into several layers instead of putting all of them into one layer as the conventional fuzzy system. The main contributions of this study can be list as follow: (1) The design of multilayer structure to improve the learning ability and flexibility of the interval type-2 fuzzy network; (2) The design of the self-evolving algorithm can help the network automatically construct its structure with suitable membership functions and rules; (3) The 3-DGMFs are used to better scope with system uncertainties and external disturbances.

The rest of this paper is organized as follows. Section II introduces the problem formulation. Section III provides structure of the SMIT2FC. Section IV presents the numerical simulation and discussion of the 5-D hyperchaotic synchronization system. Finally, Section V draws the conclusions of this study.

II. FORMULATION OF 5-D LORENZ HYPERCHAOTIC SYSTEM

According to [50], the differential equations of 5-D Lorenz hyperchaotic system are:

\[
\begin{align*}
\dot{x}_1(t) &= \chi (x_2(t) - x_1(t)) + x_4(t) \\
\dot{x}_2(t) &= \varphi x_1(t) - x_1(t)x_3(t) + x_3(t) \\
\dot{x}_3(t) &= -\gamma x_3(t) + x_1(t)x_2(t) \\
\dot{x}_4(t) &= -\varepsilon_2 x_3(t) - x_1(t)x_3(t) \\
\dot{x}_5(t) &= -\varepsilon_3 x_3(t) - \varepsilon_3 x_3(t)
\end{align*}
\]  

(1)

\[
\begin{align*}
\dot{y}_1(t) &= \chi (y_2(t) - y_1(t)) + y_4(t) + \beta(t) + \Delta f(y_1) + u(t) \\
\dot{y}_2(t) &= \varphi y_1(t) - y_1(t)y_3(t) + y_3(t) + \beta(t) + \Delta f(y_2) + u(t) \\
\dot{y}_3(t) &= -\gamma y_3(t) + y_1(t)y_2(t) + \beta(t) + \Delta f(y_3) + u(t) \\
\dot{y}_4(t) &= -\varepsilon_2 y_3(t) - y_1(t)y_3(t) + \beta(t) + \Delta f(y_4) + u(t) \\
\dot{y}_5(t) &= -\varepsilon_3 y_3(t) - \varepsilon_3 y_3(t) + \beta(t) + \Delta f(y_5) + u(t)
\end{align*}
\]  

(2)

where \( \chi, \varphi, \gamma, \beta, \) and \( \varepsilon_2, \varepsilon_3 \) respectively are the parameters for determining the behaviors and bifurcations of the hyperchaotic system, \( \chi, \varphi, \) and \( \beta \) are nonzero parameters; \( \Delta f = [\Delta f(y_1), \Delta f(y_2), \Delta f(y_3), \Delta f(y_4), \Delta f(y_5)] \) presents for the system uncertainty, which may come from the unknown nonlinear characteristics of the systems, the internal disturbances, and the measurement precision.

Defining the tracking error vector as

\[
e(t) = [e_1(t), e_2(t), e_3(t), e_4(t), e_5(t)]
\]

(3)

Obtaining error the dynamics as follow:

\[
\begin{align*}
\dot{e}_1(t) &= \chi (e_2(t) - e_1(t)) + e_1(t) + \beta(t) + \Delta f(y_1) + u(t) \\
\dot{e}_2(t) &= \varphi e_1(t) - e_1(t)e_3(t) + e_3(t) + \beta(t) + \Delta f(y_2) + u(t) \\
\dot{e}_3(t) &= -\gamma e_3(t) + e_1(t)e_2(t) + \beta(t) + \Delta f(y_3) + u(t) \\
\dot{e}_4(t) &= -\varepsilon_2 e_3(t) - e_1(t)e_3(t) + \beta(t) + \Delta f(y_4) + u(t) \\
\dot{e}_5(t) &= -\varepsilon_3 e_3(t) - \varepsilon_3 e_3(t) + \beta(t) + \Delta f(y_5) + u(t)
\end{align*}
\]  

(4)

Then (4) can be rewritten in the vector form as follow:

\[
\dot{e}(t) = Ke(t) + \mathcal{G}(t) + \Delta f(t) + u(t)
\]

(5)

where

\[
K = \begin{bmatrix}
-\chi & \chi & 0 & 1 & 0 \\
\varphi & -\gamma & 0 & 0 & 0 \\
x_2 & y_1 & -\gamma & 0 & 0 \\
-\gamma & 0 & -x_1 & -\varepsilon_2 & 0 \\
-\varepsilon_2 & -\varepsilon_3 & 0 & 0 & 0
\end{bmatrix}
\]

From (5), the ideal control vector can be given as:

\[
u^*(t) = -Ke(t) - \mathcal{G}(t) - \Delta f(t) - \dot{e}(t)
\]

(6)
However, the terms $\mathcal{G}(t)$ and $\Delta f(t)$ usually cannot be precisely known. Hence, this study proposed the SMIT2FC synchronizer to approach the ideal control signals in (6).

III. STRUCTURE OF THE SMIT2FC

A. THE STRUCTURE OF THE SMIT2FC

The $l^{th}$ rule of the SMIT2FC is described as below:

$$\text{Rule } \ell: \text{IF } (i_1, i_2) \text{ is } \lambda_{i_1,l} \text{ and } (i_2, i_3) \text{ is } \lambda_{i_2,l} \ldots \text{ and } (i_m, i_n) \text{ is } \lambda_{i_n,l} \text{ THEN } \hat{y} = \beta_{\lambda_{i_1,l}} \beta_{\lambda_{i_2,l}} \ldots \beta_{\lambda_{i_n,l}}$$

(6)

where $I = [i_1, i_2, \ldots, i_n]^T \in \mathbb{R}^n$ and $\hat{I} = [\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_n]^T \in \mathbb{R}^n$ respectively are the vector input and its derivative value; $\lambda_{ij,k}$ is input membership functions (MFs); $\beta_{\lambda_{ij,k}}$ is the fuzzy weight. $i = 1, 2, \ldots, n_i$, $j = 1, 2, \ldots, n_j$, $k = 1, 2, \ldots, n_k$, $l = 1, 2, \ldots, n_l$ and $m = 1, 2, \ldots, n_m$ respectively are the number of inputs, the number of MFs, the number of layers, the number of rules, and the number of outputs.

1) Input space:

Each node in this space presents the input variables $I = [i_1, i_2, \ldots, i_n]^T$ and $\hat{I} = [\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_n]^T$. It is directly transferred to the next space without any computation. The complexity of this step in the big-O notation is $O(1)$.

2) Membership function space:

The MFs are divided into several layers in this space. Instead of putting all MFs into one layer as the conventional fuzzy system (see Fig. 2). The 3-DGMFs (see Fig. 3) are used in this space to better handle system uncertainties and external disturbances. Then, the membership grades are calculated as follows:

FIGURE 1. The structure of the SMIT2FC using 3DMFs.

FIGURE 2. The fuzzy rule base using multilayer 3DMFs.

FIGURE 3. The illustrate of 3DMFs.
\[
\begin{align*}
\lambda_{ijk} &= \exp \left( -\frac{1}{2} \left( i - m_{ijk} \right)^2 + \left( i - m_{ijk} \right)^2 \right) \\
\tilde{\lambda}_{ijk} &= \exp \left( -\frac{1}{2} \left( i - m_{ijk} \right)^2 + \left( i - m_{ijk} \right)^2 \right)
\end{align*}
\]

where \( \lambda_{ijk} \) and \( \tilde{\lambda}_{ijk} \) are the upper and lower membership grades, respectively; \( \sigma_{ijk} \) and \( \tilde{\sigma}_{ijk} \) and \( m_{ijk} \) are the variances and the mean of the T2GMFs, respectively. The complexity of this step in the big-O notation is \( O(n_i \times n_j \times n_k) \).

3) Fuzzy firing space:

In this space, the fuzzy firing strength is calculated by using the corresponding MFs as follow:

\[
\begin{align*}
\bar{\phi}_{kl} &= \prod_{i=1}^{n_j} \lambda_{ijk} \\
\lambda_{ijk} &= \prod_{i=1}^{n_j} \lambda_{ijk}
\end{align*}
\]

The complexity of this step in the big-O notation is \( O(n_j \times n_k) \).

4) Fuzzy weight space:

The fuzzy weight is denoted as:

\[
\begin{align*}
\bar{\beta}_{jk} &= \left[ \bar{\beta}_{11} \cdots \bar{\beta}_{1n_k}; \bar{\beta}_{21} \cdots \bar{\beta}_{2n_k}; \cdots; \bar{\beta}_{n_1} \cdots \bar{\beta}_{n_{n_j}} \right] \\
\bar{\beta}_{jk} &= \left[ \bar{\beta}_{11} \cdots \bar{\beta}_{1n_k}; \bar{\beta}_{21} \cdots \bar{\beta}_{2n_k}; \cdots; \bar{\beta}_{n_1} \cdots \bar{\beta}_{n_{n_j}} \right]
\end{align*}
\]

\[
\begin{align*}
\bar{\beta}_{jk} &= \left[ \bar{\beta}_{11} \cdots \bar{\beta}_{1n_k}; \bar{\beta}_{21} \cdots \bar{\beta}_{2n_k}; \cdots; \bar{\beta}_{n_1} \cdots \bar{\beta}_{n_{n_j}} \right] \\
\bar{\beta}_{jk} &= \left[ \bar{\beta}_{11} \cdots \bar{\beta}_{1n_k}; \bar{\beta}_{21} \cdots \bar{\beta}_{2n_k}; \cdots; \bar{\beta}_{n_1} \cdots \bar{\beta}_{n_{n_j}} \right]
\end{align*}
\]

The complexity of this step in the big-O notation is \( O(1) \).

5) Output space:

The \( k \)-th output of SMIT2FC is represented as:

\[
u_{SMIT2FC}^m = o_m = \frac{1}{2} \left( \sum_{k=1}^{n_j} \sum_{i=1}^{n_i} (\bar{\phi}_{ik} \bar{\beta}_{klm}) + \sum_{k=1}^{n_j} \sum_{i=1}^{n_i} (\phi_{ik} \beta_{klm}) \right)
\]

Choosing the Lyapunov cost function as

\[
E = \frac{1}{2} (e_m(t))^2,
\]

where \( m=1,2,...,5 \). The adaptive laws for adjusting the proposed network’s parameters can be calculated through chain rule as:

Using the gradient descent method, the online adaptive laws for updating the proposed controller’s parameters are given as:

\[
\begin{align*}
\dot{\beta}_{klm}(t+1) &= \dot{\beta}_{klm}(t) - \eta_{\beta} \frac{\partial E(t)}{\partial \beta_{klm}} \\
&= \dot{\beta}_{klm}(t) - \frac{1}{2} \eta_{\beta} e_m(t) \frac{\partial \bar{\phi}_{kl}}{\partial \beta_{klm}} \\
\dot{\lambda}_{ijk}(t+1) &= \dot{\lambda}_{ijk}(t) - \lambda_{ijk} \frac{\partial E(t)}{\partial \lambda_{ijk}} \\
&= \dot{\lambda}_{ijk}(t) - \frac{1}{2} \eta_{\lambda} e_m(t) \frac{\partial \tilde{\lambda}_{ijk}}{\partial \lambda_{ijk}} \\
\dot{\mu}_{ijk}(t+1) &= \dot{\mu}_{ijk}(t) - \mu_{ijk} \frac{\partial E(t)}{\partial \mu_{ijk}} \\
&= \dot{\mu}_{ijk}(t) - \frac{1}{2} \eta_{\mu} e_m(t) \frac{\partial \mu_{ijk}}{\partial \mu_{ijk}}
\end{align*}
\]
where \( \hat{\eta}_\beta, \hat{\eta}_m, \hat{\eta}_\sigma \) are the positive learning rates.

The total complexity of steps in big-O notation is given as:

\[
L = O(1) + O(n_t \times n_k) + O(n_t \times n_k) + O(1) + O(n_k \times n_m) \tag{19}
\]

Proof: Considering the Lyapunov cost function below:

\[
V(t) = \frac{1}{2} \left( e_m(t) \right)^2 \tag{20}
\]

Thus,

\[
\Delta V(t) = V(t+1) - V(t) = \frac{1}{2} \left[ \left( e_m(t+1) \right)^2 - \left( e_m(t) \right)^2 \right] \tag{21}
\]

Using the Taylor expansion for \( \hat{\beta}_{\text{kin}} \), obtains

\[
e_m(t+1) = e_m(t) + \Delta e_m(t) \equiv e_m(t) + \left[ \frac{\partial e_m(t)}{\partial \hat{\beta}_{\text{kin}}} \right] \Delta \hat{\beta}_{\text{kin}} \tag{22}
\]

where \( \Delta \hat{\beta}_{\text{kin}} = \hat{\beta}_{\text{kin}}(k+1) - \hat{\beta}_{\text{kin}}(k) \)

From (14),

\[
\frac{\partial e_m(t)}{\partial \hat{\beta}_{\text{kin}}} = \frac{1}{2} \frac{1}{\phi_{ki}} \tag{23}
\]

Rewrite (22) using (23) and (14), obtains

\[
e_m(t+1) = e_m(t) - \left( \frac{1}{2} \frac{1}{\phi_{ki}} \right) \frac{1}{2} \left( \hat{\beta}_{\text{kin}} e_m(t) \phi_{ki} \right)
\]

\[
= e_m(t) \left[ 1 - \hat{\eta}_m \left( \frac{1}{2} \frac{1}{\phi_{ki}} \right)^2 \right] \tag{24}
\]

Rewrite (21) using (24), obtains

\[
\Delta V(t) = \frac{1}{2} \left( e_m(t) \right)^2 \left[ 1 - \hat{\eta}_m \left( \frac{1}{2} \frac{1}{\phi_{ki}} \right)^2 \right] - 1
\]

\[
= \frac{1}{2} \left( e_m(t) \right)^2 \left[ \hat{\eta}_m \left( \frac{1}{2} \frac{1}{\phi_{ki}} \right)^2 - 2 \hat{\eta}_m \left( \frac{1}{2} \frac{1}{\phi_{ki}} \right)^2 \right]
\]

\[
= \frac{1}{2} \hat{\eta}_m \left( e_m(t) \right)^2 \left( \frac{1}{2} \frac{1}{\phi_{ki}} \right)^2 \left( \hat{\eta}_m \left( \frac{1}{2} \frac{1}{\phi_{ki}} \right)^2 - 2 \right) \tag{25}
\]

From (25), if \( \hat{\eta}_m \) is chosen to satisfy \( 0 < \hat{\eta}_m < \frac{8}{\left( \frac{1}{\phi_{ki}} \right)^2} \), then

\[
\Delta V(t) < 0
\]

Therefore, the convergence of the designed adaptation laws is guaranteed according to the Lyapunov theorem. Similarly, the convergence proof for \( \hat{\eta}_m \) and \( \hat{\eta}_\sigma \) can be obtained.

### B. STRUCTURE LEARNING ALGORITHM

In this section, the self-evolving algorithm is designed to construct the network structure of the proposed SMIT2FC synchronizer autonomously. The procedure of online self-evolving algorithm is given as follow:

**Step 1:** Initialize the structure of the SMIT2FC network with a few membership functions and initial corresponding rules.

**Step 2:** Evaluate the maximum contribution of the membership functions by considering its membership grade. The evaluation criteria for generating a new membership function is given as:

\[
\lambda_i^m < \lambda_i^d \tag{26}
\]

\[
\lambda_{\max}^i = \max \left[ \lambda_{ij1}, \lambda_{ij2}, \ldots, \lambda_{ijn_i} \right] \quad \lambda_i^d = \frac{\lambda_{ij1} + \lambda_{ijk}}{2} \tag{27}
\]

where \( \lambda_i^d \) is the predefined generating threshold; \( \lambda_{\max}^i \) is the maximum membership grade for the \( i \)-th input.

If the criteria in (26) are satisfied, the initial parameters of the new membership function are set by

\[
m_{ij(k+1)} = i(t) \tag{28}
\]

\[
\left[ \sigma_{ij(k+1)}^s, \sigma_{ij(k+1)}^g \right] = \left[ \left( \sigma_{ij}^s - \Delta \sigma \right), \left( \sigma_{ij}^s + \Delta \sigma \right) \right] \tag{29}
\]

where \( \sigma_{ij}^s \) and \( \Delta \sigma \) are the initial value and the uncertainty term for the variance of the new Gaussian membership function.

**Step 3:** Evaluate the minimum contribution of the membership functions by considering its membership grade. The evaluation criteria for deleting a less contribution membership function is given as:

\[
\lambda_{\min}^i < \lambda_i^d \tag{30}
\]

\[
\lambda_{\min}^i = \min \left[ \lambda_{ij1}, \lambda_{ij2}, \ldots, \lambda_{ijn_i} \right] \tag{31}
\]

where \( \lambda_i^d \) is the predefined deleting threshold; \( \lambda_{\min}^i \) is the minimum membership grade for the \( i \)-th input.

If the criteria in (26) are satisfied, the less contribution membership function will be deleted.

**Step 4:** When the next state of \( \hat{I}_k \) arrives, go to Step 2. Then, repeat until the synchronization is terminated.

By applying the proposed self-evolving algorithm, the SMIT2FC synchronizer can obtain a suitable structure with suitable membership functions and rules.

The illustration for the operation of the self-evolving algorithm is given in Figs. 4 and 5. Figure 4a shows that when the current layers do not better cover the new input value, and the maximum membership grade for this input is smaller than the predefined generating threshold, then a new layer and new rules should be added as Fig. 4b. Figure 5a shows that when the new input value is far from one layer, and the minimum membership grade for this layer is smaller than the predefined deleting threshold, this inessential layer and its rules should be deleted as Fig. 5b.

At the beginning of the control process, it only needs to design a few layers with some initial membership functions and rules. After that, the self-evolving algorithm can add a new essential layer or delete an inessential layer. Therefore, the network structure of the proposed method can automatically construct by using the self-evolving algorithm.
VOL. XX, 2017

IV. SIMULATION RESULTS

In this section, the validity of the proposed SMIT2FC controller is verified by conducting two cases simulation of synchronization hyperchaotic system. The initial parameters for the SMIT2FC controller are chosen as \( \sigma_{\text{init}} = 0.3 \), \( \Delta \sigma = 0.05 \), \( n_i = 5 \), \( n_j = 3 \), \( n_k = 2 \), \( \lambda_i = 0.1 \), \( \lambda_d = 0.01 \). The initial conditions for the master and slave system are respectively set as

\[
\begin{align*}
\mathbf{x}_M \bigg|_{t=0} &= [20, 20, 20, 20, 20]^T, \\
\mathbf{x}_S \bigg|_{t=0} &= [0, 0, 0, 0, 0]^T.
\end{align*}
\]

The external disturbances is chosen as

\[
\begin{align*}
\mathbf{d}(t) &= \begin{bmatrix} 0.2 \cos \pi t, 0.2 \cos \pi t, 0.2 \cos \pi t, 0.2 \cos \pi t, 0.2 \cos \pi t \end{bmatrix}^T.
\end{align*}
\]

The system uncertainties is chosen as

\[
\begin{align*}
\mathbf{r}(t) &= \begin{bmatrix} 0.1 \sin \pi t, 0.1 \sin \pi t, 0.1 \sin \pi t, 0.1 \sin \pi t, 0.1 \sin \pi t \end{bmatrix}^T.
\end{align*}
\]

The scheme for the hyperchaotic synchronization system using the proposed the SMIT2FC controller is shown in Fig. 6.

The proposed controller \( u_{\text{SMIT2FC}}^m \), in (13) is then applied to synchronize the slave hyperchaotic system in (2). The tracking error vector in (3) is used as the input space of the SMIT2FC controller. The membership grades in (7), (8) and conditions (26), (30) are used to determine whether a new layer should be added or delete an inessential layer. The adaptive laws (14)-(18) are used to update the network’s parameters online.

FIGURE 6. The scheme for the SMIT2FC hyperchaotic synchronization system.

**Case 1:**

In this case, the parameters for determining the behaviors and bifurcations of the 5-D hyperchaotic systems are set by \( a = 10, b = \frac{8}{3}, c = 28, h_1 = -2, h_2 = 0.4, h_3 = 8 \).

The simulation results in Figs 7 – 11 have demonstrated the effectiveness of the proposed method. Figure. 7 shows the synchronized trajectories of master and slave hyperchaotic systems in three dimensions. Figure. 8 shows the synchronized trajectories of each element in the time domain. The control signals and the tracking errors for synchronization of the 5-D hyperchaotic systems are shown in Fig. 9 and Fig. 10, respectively. The change in number of layers using the self-evolving algorithm is shown in Fig. 11.

The above simulation results show that the proposed controller can quickly synchronize the 5-D hyperchaotic system fastly with minor tracking errors. The root mean square error (RMSE) in this case is 2.257.
FIGURE 7. The synchronized trajectories of the 5-D hyperchaotic systems in three dimension.

FIGURE 8. The synchronized trajectories of the 5-D hyperchaotic systems in the time domain.

FIGURE 9. The control signals for synchronization of the 5-D hyperchaotic systems.
Case 2:

In this case, the parameters for determining the behaviors and bifurcations of the 5-D hyperchaotic systems are set by

\[ a = 10, \quad b = \frac{8}{3}, \quad c = 28, \quad h_1 = -2.5, \quad h_2 = -0.12, \quad h_3 = 11.3. \]

The simulation results in Figs 12–16 have demonstrated the effectiveness of the proposed method. Figure 12 shows the synchronized trajectories of master and slave hyperchaotic systems in three dimension. Figure 13 shows the synchronized trajectories of each element in time domain. The control signals and the tracking errors for synchronization of the 5-D hyperchaotic systems are shown in Fig. 14 and Fig. 15, respectively. The change in number of layers using the self-evolving algorithm is shown in Fig. 16. The above simulation results show that the proposed controller can quickly synchronize the 5-D hyperchaotic system fastly with minor tracking errors. The root mean square error (RMSE) in this case is 2.261.
The results show that the proposed SMIT2FC controller can fastly synchronize the master and slave hyperchaotic synchronization systems in both cases. Thus, the synchronization errors between them quickly converge to zero. The self-evolving algorithm can automatically adjust the number of layers to achieve a suitable structure during the synchronization process. In summary, the proposed controller has strong robustness since the RMSE varies in a small range, even under the influence of external disturbances and system uncertainties.

In the self-developed algorithm, the choice of predefined thresholds for adding or removing layers has dramatically influenced the efficiency of the synchronization performance process. If the generating
threshold is too large, it will be challenging to create new layers. On the contrary, if the generating threshold is too small, it will create many further layers, and the computation cost will be high. Similarly, if the generating threshold is too small, it will rarely delete an insessional layer. On the contrary, If the deletion threshold is too large, it will delete most of the layers, and result in the remaining layers not covering the inputs well. In this study, we used the trial-and-error method to achieve suitable thresholds.

V. CONCLUSION

In this study, the design and implementation of SMIT2FC have been proposed to synchronize the hyperchaotic system. The multilayer structure is applied to enhance the learning ability and flexibility of the proposed network. The adaptive online laws for adjusting the network’s parameters are derived by the gradient descent method. The network structure can automatically construct by using the self-evolving algorithm. The 3-DGMFs are applied to handle system uncertainties and external disturbances better. The efficiency and performance of the designed network are validated by numerical simulations on the synchronization of hyperchaotic systems. Apply some meta-heuristic algorithms to optimal predefined thresholds in the self-evolving algorithm will be our future work.

REFERENCES

[1] L. A. Zadeh, “Fuzzy sets” Inf. Control, vol. 8, pp. 338–353, 1965.
[2] Y. Chang, Y. Wang, F. E. Alsaadi, and G. Zong, “Adaptive fuzzy output-feedback tracking control for switched stochastic pure-feedback nonlinear systems,” International Journal of Adaptive Control and Signal Processing, vol. 33, no. 10, pp. 1567-1582, 2019.
[3] S. Farajadjan, and S. H. Hosseini, “Design of an optimal fuzzy controller to obtain maximum power in solar power generation system,” Solar Energy, vol. 182, pp. 161-178, 2019.
[4] L. G. Marin, N. Cruz, D. Sáez, M. Sumner, and A. Núñez, “Prediction interval methodology based on fuzzy numbers and its extension to fuzzy systems and neural networks,” Expert Systems with Applications, vol. 119, pp. 128-141, 2019.
[5] S. E. El-Khamy, N. O. Korany, and A. G. Mohamed, “A new fuzzy-DNA image encryption and steganography technique,” IEEE Access, vol. 8, pp. 148935-148951, 2020.
[6] M. H. Homaei, F. Soleimani, S. Shamsibhand, A. Mosavi, N. Nabipour, and A. R. Varkonyi-Kozy, “An enhanced distributed congestion control method for classical 6LowPAN protocols using fuzzy decision system,” IEEE Access, vol. 8, pp. 20628-20645, 2020.
[7] C.-L. Zhang, X.-Z. Wu, and J. Xu, “Particle Swarm Sliding Mode-Fuzzy PID Control Based on Maglev System,” IEEE Access, vol. 9, pp. 96337-96344, 2021.
[8] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning—I,” Information Sciences, vol. 8, no. 3, pp. 199-249, 1975.
[9] H. Hassani, J. Zarei, M. M. Arefi, and R. Razavi-Far, “r2Slices-based general type-2 fuzzy fusion of support vector machines with application to bearing fault detection,” IEEE Transactions on Industrial Electronics, vol. 64, no. 9, pp. 7210-7217, 2017.
[10] A. Mohammadzadeh, and E. Kayacan, “A novel fractional-order type-2 fuzzy control method for online frequency regulation in ac microgrid,” Engineering Applications of Artificial Intelligence, vol. 90, pp. 103483, 2020.
[11] Q. Shen, Y. Shi, R. Jia, and P. Shi, “Design on Type-2 Fuzzy-based Distributed Supervisory Control with Backlash-like Hysteresis,” IEEE Transactions on Fuzzy Systems, 2020.

[12] O. Carvajal, P. Melin, I. Miramontes, and G. Prado-Arechiga, “Optimal design of a general type-2 fuzzy classifier for the pulse level and its hardware implementation,” Engineering Applications of Artificial Intelligence, vol. 97, pp. 104069, 2021.
[13] Y. Zhao, J. Wang, F. Yan, and Y. Shen, “Adaptive sliding mode fault-tolerant control for type-2 fuzzy systems with distributed delays,” Information Sciences, vol. 473, pp. 227-238, 2019.
[14] F. Ali, S. R. Islam, D. Kwak, P. Khan, N. Ullah, S.-J. Yoo, and K. S. Kwak, “Type-2 fuzzy ontology–aided recommendation systems for IoT–based healthcare,” Computer Communications, vol. 119, pp. 138-155, 2018.
[15] E. Ontiveros, P. Melin, and O. Castillo, “High order α-planes integration: a new approach to computational cost reduction of general type-2 fuzzy systems,” Engineering Applications of Artificial Intelligence, vol. 74, pp. 186-197, 2018.
[16] O. Castillo, and L. Amador-Angulo, “A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design,” Information Sciences, vol. 460, pp. 476-496, 2018.
[17] Q. Liang, and J. M. Mendel, “Interval type-2 fuzzy logic systems: theory and design,” IEEE Transactions on Fuzzy Systems, vol. 8, no. 4, pp. 535-550, 2000.
[18] X. Ma, P. Wu, L. Zhou, H. Chen, T. Zheng, and J. Ge, “Approaches based on interval type-2 fuzzy aggregation operators for multiple attribute group decision making,” International Journal of Fuzzy Systems, vol. 18, no. 4, pp. 697-715, 2016.
[19] P. A. Herman, G. Prasad, and T. M. McGinnity, “Designing an Interval Type-2 Fuzzy Logic System for Handling Uncertainty Effects in Brain–Computer Interface Classification of Motor Imagery Induced EEG Patterns,” IEEE Transactions on Fuzzy Systems, vol. 25, no. 1, pp. 29-42, 2017.
[20] T.-L. Le, “Self-Organizing Recurrent Interval Type-2 Petri Fuzzy Design for Time-Varying Delay Systems,” IEEE Access, vol. 7, pp. 10505-10514, 2018.
[21] Z. Du, Y. Kao, H. R. Karimi, and X. Zhao, “Interval Type-2 Fuzzy Sampled-Data Sliding Mode Control for Nonlinear Unreliable Networked Control Systems,” IEEE Transactions on Fuzzy Systems, 2019.
[22] T.-L. Le, T.-T. Huynh, L.-Y. Lin, C.-M. Lin, and F. Chao, “A k-means interval type-2 fuzzy neural network for medical diagnosis,” International Journal of Fuzzy Systems, vol. 21, no. 7, pp. 2258-2269, 2019.
[23] A. Dey, A. Pal, and H. V. Long, “Fuzzy minimum spanning tree with interval type 2 fuzzy arc length formulation and a new genetic algorithm,” Soft Computing, vol. 24, no. 6, pp. 3963-3974, 2020.
[24] R. Kavikumar, R. Sakhthivel, O.-M. Kwon, and B. Kaviarasan, “Robust model reference tracking control for interval type-2 fuzzy stochastic systems,” IET Control Theory & Applications, vol. 14, no. 9, pp. 1123-1134, 2020.
[25] S. Karagöz, M. Deveci, V. Simic, and N. Aydin, “Interval type-2 Fuzzy ARAS method for recycling facility location problems,” Journal of Intelligent & Fuzzy Systems, vol. 32, no. 6, pp. 3981-3991, 2017.
[26] S. B. Pandu, C. Sundarabanal, S. Nrinath, T. S. Krishnan, G. S. Priya, C. Balasundar, J. Sharma, G. Soundarya, P. Siano, and H. H. Alhelou, “Power Quality Enhancement in Sensitive Local Distribution Grid Using Interval Type-II Fuzzy Logic Controlled DSTATCOM,” IEEE Access, vol. 9, pp. 59888-59899, 2021.
[27] D. Wu, “On the fundamental differences between interval type-2 and type-1 fuzzy logic controllers,” IEEE Transactions on Fuzzy Systems, vol. 20, no. 5, pp. 832-848, 2012.
[28] S. Türk, M. Deveci, E. Özcan, F. Cantez, and R. John, “Interval type-2 fuzzy sets improved by Simulated Annealing for locating the electric charging stations,” Information Sciences, vol. 547, pp. 641-666, 2021.
[29] C.-M. Lin, T.-L. Le, and T.-T. Huynh, “Self-evolving function-link interval type-2 fuzzy neural network for nonlinear system identification and control,” Neurocomputing, vol. 275, pp. 2239-2250, 2018.
[30] P. Ganesan, S. J. Rathna, and R. Sairud, “Application of artificial neural network to map the performance characteristics of boiler using different algorithms,” International Journal of Green Energy, pp. 1-13. 2021.
C.-F. Juang, R.-B. Huang, and Y.-Y. Lin, “A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing,” IEEE Transactions on Fuzzy Systems, vol. 17, no. 5, pp. 1092-1105, 2009.

Y.-Y. Lin, J.-Y. Chang, and C.-T. Lin, “A TSK-type-based self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) and its applications,” IEEE Transactions on Industrial Electronics, vol. 61, no. 1, pp. 447-459, 2014.

S.-Y. Chen and T.-S. Liu, “Intelligent tracking control of a PMLSM using self-evolving probabilistic fuzzy neural network,” IET Electric Power Applications, vol. 11, no. 6, pp. 1043-1054, 2017.

T.-L. Le, C.-M. Lin, and T.-T. Huynh, “Self-evolving type-2 fuzzy brain emotional learning control design for chaotic systems using PSO,” Applied Soft Computing, vol. 73, pp. 418-433, 2018.

H.-G. Han, Z.-Y. Chen, H.-X. Liu, and J.-F. Qiao, “A self-organizing interval type-2 fuzzy-neural-network for modeling nonlinear systems,” Neurocomputing, vol. 290, pp. 196-207, 2018.

T.-L. Le, T.-T. Huynh, and C.-M. Lin, “Self-evolving interval type-2 wavelet cerebellar model articulation control design for uncertain nonlinear systems using PSO,” International Journal of Fuzzy Systems, vol. 21, no. 8, pp. 2524-2541, 2019.

T. Zhao, P. Li, and J. Cao, “Soft sensor modeling of chemical process based on self-organizing recurrent interval type-2 fuzzy neural network,” ISA transactions, vol. 84, pp. 237-246, 2019.

A. Salimi-Badr and M. M. Ebadzadeh, “A novel self-organizing fuzzy neural network to learn and mimic habitual sequential tasks,” IEEE transactions on cybernetics, 2020. DOI: 10.1109/TCYB.2020.2984646

D. Ge and X.-J. Zeng, “Learning data streams online—An evolving fuzzy system approach with self-learning/adaptive thresholds,” Information Sciences, vol. 507, pp. 172-184, 2020.

X. Gu, P. Angelov, and Z. Zhao, “Self-organizing fuzzy inference ensemble system for big streaming data classification,” Knowledge-Based Systems, vol. 218, pp. 106870, 2021.

Z. Wang and J. Fei, “Fractional-Order Terminal Sliding Mode Control Using Self-Evolving Recurrent Chebyshev Fuzzy Neural Network for MEMS Gyroscope,” IEEE Transactions on Fuzzy Systems, 2021. DOI: 10.1109/TFUZZ.2021.3094717

J. Shah and W. J. I. t. Wang, “An evolving neuro-fuzzy classifier for fault diagnosis of gear systems,” ISA Transactions, 2021. https://doJ.10.1016/j.isatra.2021.05.019

Ö. Atan, F. Kutlu, and O. Castillo, “Intuitionistic fuzzy sliding controller for uncertain hyperchaotic synchronization,” International Journal of Fuzzy Systems, vol. 22, pp. 1430-1443, 2020.

Q. Yang, C. Chen, “A 5D hyperchaotic system with three positive Lyapunov exponents coined,” International Journal of Bifurcation and Chaos, vol. 23, no. 06, pp. 1350109, 2013.

S. Vaidyanathan, "Adaptive anti-synchronization of 5-D novel hyperchaotic systems and SPICE implementation." IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, pp. 1-7, 2014.

A. Mohammadzadeh and S. Ghaemi, “Synchronization of uncertain fractional-order hyperchaotic systems by using a new self-evolving non-singleton type-2 fuzzy neural network and its application to secure communication,” Nonlinear Dynamics, vol. 88, no. 1, pp. 1-19, 2017.

S. He, K. Sun, H. Wang, X. Mei, and Y. Sun, “Generalized synchronization of fractional-order hyperchaotic systems and its DSP implementation,” Nonlinear Dynamics, vol. 92, no. 1, pp. 85-96, 2018.

A. Ouannas, S. Bendoukha, C. Volos, N. Boumaza, and A. Karouma, Automation, “Synchronization of fractional hyperchaotic Rabinovich systems via linear and nonlinear control with an application to secure communications,” International Journal of Control, Automation and Systems, vol. 17, no. 9, pp. 2211-2219, 2019.

C. Zhou, C. Yang, D. Xu, and C.-Y. Chen, “Dynamic analysis and finite-time synchronization of a new hyperchaotic system with coexisting attractors,” IEEE Access, vol. 7, pp. 52896-52902, 2019.

T. Sangpet and S. Kuntanapreeda, “Finite-time synchronization of hyperchaotic systems based on feedback passivation,” Chaos, Solitons & Fractals, vol. 132, pp. 109605, 2020.