The Synchronization of Hyperchaotic Systems Using a Novel Interval Type-2 Fuzzy Neural Network Controller

TIEN-LOC LE*, (Member, IEEE), AND VAN-BINH NGO
Faculty of Mechatronics and Electronics, Lac Hong University, Bien Hoa, Dong Nai 810000, Vietnam
Corresponding author: Tien-Loc Le (tienloc@lhu.edu.vn)
This work was supported by the Lac Hong University, Bien Hoa, Dong Nai, Vietnam.

ABSTRACT This paper proposed a novel interval type-2 fuzzy neural network controller (NT2FC) to synchronize 5-D hyperchaotic systems with noise disturbance and system uncertainties. In the proposed controller, the type 2 fuzzy set is designed with the 3-dimensional Gaussian membership functions (3DGMFs) to increase the system’s ability to respond to uncertainty. The parameters of the NT2FC controller are updated online via adaptation laws, which are built based on the gradient descent approach. The system stability is ensured through the Lyapunov stability analysis. In addition, the modified Jaya algorithm (MJA) is applied to optimize the learning rates in adaptation laws. Finally, the efficiency of the proposed NT2FC is examined by the numerical simulation of the hyperchaotic system’s synchronization.

INDEX TERMS 5-D hyperchaotic systems, fuzzy neural network, type-2 fuzzy system, 3DGMFs, Jaya algorithm.

I. INTRODUCTION
In recent years, the chaotic system’s synchronization has been studied and applied in many different fields because of its unique dynamic characteristics including unpredictable behaviours, nonlinear and sensitive response to initial conditions [1]. Chaotic system’s synchronization is a procedure that generates the control force to make one or more slave chaotic systems follow the behavior of the master chaotic system. Some notable synchronization methods of the chaotic system have been proposed in [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], and [12]. In 2013, Lin et al. proposed a wavelet cerebellar model articulation controller for synchronization of the unified chaotic system [2]. In 2018, Mufti et al. presented a synchronization and antisynchronization using sliding mode control [3]. In 2021, Dai et al. introduced a robust synchronization of chaotic systems using a neural network approach [10]. Hyperchaotic systems are high-dimensional chaotic systems, which have at least two positives and one negative Lyapunov exponents to ensure the boundedness of the systems [13], [14]. Therefore it has more complicated topological structures and unpredictable dynamical behaviour than the typical chaotic system. The synchronization of 5-D hyperchaotic systems has been developed in [13], [15], [16], [17], and [18], which considers five dimensions space. Nowadays, along with the development of synchronous control techniques, the hyperchaotic system is widely applied in various fields, especially in signal processing and information security [19].

Recently, the combination of neural network (NN) and fuzzy logic system (FLS) has produced the fuzzy neural network (FNN). Therefore, FNN has the reasoning approach of FLS and the learning capability of NN. The traditional fuzzy logic systems are also known as the type-1 fuzzy logic system (T1FLS), which use the type-1 membership functions in the consequent and the antecedent space. Some notable studies using T1FLS have been introduced in [20], [21], [22], and [23]. In 2012, Faieghi et al. proposed a fuzzy sliding mode controller for controlling the fractional-order Liu system [20]. In 2015, Delavari and Shokriani presented a fuzzy modeling method for synchronization of the hyperchaotic complex system [21]. Recently, some studies have shown that the type-2 fuzzy logic system (T2FLS) can better deal with the uncertainties than the T1FLS [24], [25], [26], [27]. However,
the computation cost in the T2FLS is often enormous [28]. Therefore, the interval type-2 fuzzy logic system (IT2FLS) is introduced in [29] to reduce the computational complexity. Since then, the IT2FLS has been applied in many different fields [30], [31], [32], [33], [34], [35], [36], [37]. In 2018, Jiang et al. presented an IT2FLS for stock index forecasting [31]. In 2017, Mohadeszadeh and Delavari proposed a type-2 fuzzy sliding mode controller for synchronization of hyper-chaotic systems [37]. This paper proposed a novel IT2FLS with 3-D type-2 Gaussian membership functions to achieve better synchronization performance of the chaotic system’s synchronization.

Choosing the learning rates is very important in the adaptation law for updating the network’s parameters. It can make the learning process oscillate or even not converge [38]. Most studies in the literature often use trial and error to obtain the suitable learning rates, but this method is often time-consuming and difficult to achieve optimal coefficients [39]. Therefore, we applied the MJA to obtain the optimal learning rates for the designed adaptation law in this study. To enhance and extend the searchability of the traditional Jaya algorithm (JA), our MJA additionally considers second-best and third-best solutions as well as second-worst and third-worst solutions. Moreover, random parameters are added to the updating formula, which makes the MJA algorithm more flexible than the traditional JA. Rao first introduced the original JA in 2016, which is known as the parameterless algorithm [40]. JA’s optimization mechanism is asymptotically the best solution and away from the worst solution in the swarm. Unlike most other optimization algorithms, JA does not require using any particular parameters other than two general parameters, such as the number of iterations and the population size. Some notable applications of JA for optimization problems in various fields can be found in [41], [42], [43], [44], [45], and [46]. In 2019, Yu et al. presented a Jaya algorithm to solve economic load dispatch problems [43]. In 2021, Deboucha et al. introduced a modified deterministic Jaya -based MPPT to enhance the PV system’s effectiveness [46].

Motivated by above discussion, this work provided a novel controller, which combines the advantages of the IT2FNN, the 3DGMFs, the self-evolving algorithm, and the modified model Jaya algorithm. The significant contributions of the proposed synchronization system are (1) The successful...


Considering the following 5-D Lorenz hyperchaotic system

\[ \begin{align*}
\dot{x}_1(t) &= a(y_1(t) - x_1(t)) + v_1(t) \\
\dot{y}_1(t) &= cx_1(t) - x_1(t)z_1(t) + w_1(t) \\
\dot{z}_1(t) &= -bz_1(t) + x_1(t)y_1(t) \\
\dot{v}_1(t) &= -h_1v_1(t) - x_1(t)z_1(t) \\
\dot{w}_1(t) &= -h_2x_1(t) - h_3y_1(t)
\end{align*} \]

(1)

The design of IT2FNN with parameters can be updated online using the designed adaptation laws; (2) The design of 3-D type-2 Gaussian membership functions for better cope with the system uncertainties; (3) The design of MJA for optimization the suitable learning rates.

**II. PROBLEM FORMULATION**

Considering the following 5-D Lorenz hyperchaotic system in [47] and [48] as follow:

\[ \begin{align*}
\dot{x}_2(t) &= a(y_2(t) - x_2(t)) + v_2(t) + \rho_4(t) + \Delta f(x_2) + u_c(t) \\
\dot{y}_2(t) &= cx_2(t) - x_2(t)z_2(t) + w_2(t) + \rho_3(t) + \Delta f(y_2) + u_c(t) \\
\dot{z}_2(t) &= -bz_2(t) + x_2(t)\dot{y}_2(t) + \rho_4(t) + \Delta f(z_2) + u_c(t) \\
\dot{v}_2(t) &= -h_1v_2(t) - x_2(t)\dot{z}_2(t) + \rho_4(t) + \Delta f(v_2) + u_c(t) \\
\dot{w}_2(t) &= -h_2x_2(t) - h_3y_2(t) + \rho_4(t) + \Delta f(w_2) + u_c(t)
\end{align*} \]

(2)

where \( f_1 = [\dot{x}_1(t), \dot{y}_1(t), \dot{z}_1(t), \dot{v}_1(t), \dot{w}_1(t)] \) and \( f_2 = [\dot{x}_2(t), \dot{y}_2(t), \dot{z}_2(t), \dot{v}_2(t), \dot{w}_2(t)] \) respectively are the state vectors of the master and slave hyperchaotic system; \( a, b, c, h_1, h_2 \) and \( h_3 \) are the characteristic parameters \( (a, b, h_1 \neq 0); \rho = [\rho_4(t), \rho_3(t), \rho_4(t), \rho_4(t), \rho_4(t)] \) and \( \Delta f = [\Delta f(x_2), \Delta f(y_2), \Delta f(z_2), \Delta f(v_2), \Delta f(w_2)] \) respectively are the external disturbances vector and system

**FIGURE 2.** The illustration of the 3-D Gaussian membership functions using in NT2FC.

**FIGURE 3.** The synchronization scheme using the proposed NT2FC synchronizer.
uncertainty vector; $u = [u_x(t), u_y(t), u_z(t), u_v(t), u_w(t)]$ are the synchronizer control vector.

The tracking error vector between the master and slave hyperchaotic system is denoted by $e = [e_x(t), e_y(t), e_z(t), e_v(t), e_w(t)],$ and given as follow:

$$
\begin{align*}
e_x(t) &= x_2(t) - x_1(t) \\
e_y(t) &= y_2(t) - y_1(t) \\
e_z(t) &= z_2(t) - z_1(t) \\
e_v(t) &= v_2(t) - v_1(t) \\
e_w(t) &= w_2(t) - w_1(t)
\end{align*}
$$

From (1), (2), and (3), the dynamic tracking error vector can be obtained as follow:

$$
\begin{align*}
\dot{e}_x(t) &= a(\dot{x}_x(t) - e_x(t)) + e_y(t) + \rho_x(t) + \Delta f(x_2) + u_x(t) \\
\dot{e}_y(t) &= c e_x(t) - x_2(t)z_2(t) + x_1(t)z_1(t) - e_w(t) \\
&\quad + \rho_y(t) + \Delta f(y_2) + u_y(t) \\
\dot{e}_z(t) &= -be_x(t) + y_2(t) - x_1(t)y_1(t) + \rho_z(t) \\
&\quad + \Delta f(z_2) + u_z(t) \\
\dot{e}_v(t) &= -h_1 e_x(t) - x_2(t)z_2(t) + x_1(t)z_1(t) \\
&\quad + \rho_v(t) + \Delta f(v_2) + u_v(t) \\
\dot{e}_w(t) &= -h_2 e_x(t) - h_3 e_y(t) + \rho_w(t) + \Delta f(w_2) + u_w(t) \\
\end{align*}
$$

(4)

Rewriting (4), yields

$$
\dot{e}(t) = Ae(t) + \rho(t) + \Delta f(t) + u(t)
$$

(5)

where

$$
A = \begin{bmatrix}
-a & a & 0 & 1 & 0 \\
(c - z_2) & 0 & -x_1 & 0 & -1 \\
y_1 & x_2 & -b & 0 & 0 \\
-z_2 & 0 & -x_1 & -h_1 & 0 \\
-h_2 & -h_3 & 0 & 0 & 0 \\
\end{bmatrix}
$$

From (5), the ideal control signal $u^*(t)$ are given as follows:

$$
u^*(t) = -Ae(t) - \rho(t) - \Delta f(t) + \dot{e}(t)
$$

(6)

Since $\rho(t)$ and $\Delta f(t)$ are not exactly known, thus the $u^*(t)$ in (6) is unobtainable. Therefore, the proposed NT2FC synchronizer is designed in this study to emulate the ideal synchronizer.

III. DESIGN OF NT2FC CONTROLLER

A. STRUCTURE OF NT2FC

The fuzzy inference, which represents the relationship between the consequent and the antecedent, is given as follows:

Rule $k^{th}$: IF $(x_1, \dot{x}_1)$ is $\tilde{\mu}_{1j}$ and $(x_2, \dot{x}_2)$
is $\tilde{\mu}_{2j}, \ldots$, and $(x_{n_i}, \dot{x}_{n_i})$ is $\tilde{\mu}_{n_j}$ Then $\tilde{w}_{km} = [\bar{w}_{km} \bar{w}_{km}]$

for $i = 1, 2, \ldots, n_i$; $j = 1, 2, \ldots, n_j$; $k = 1, 2, \ldots, n_k$;

$m = 1, 2, \ldots, n_m$. (7)

where $X = [x_1, x_2, \ldots, x_{n_i}]^T \in \Re^{n_i}$ and $\dot{X} = [\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_{n_i}]^T$

$\in \Re^{n_i}$ are the input vector and its derivative, respectively; $n_i$ and $n_m$ are the number of inputs and outputs, respectively; $n_k$ and $n_j$ are the number of rules and the number of membership functions in each input, respectively; $\tilde{\mu}_{ij} = \begin{bmatrix} \mu_{ij} & \bar{\mu}_{ij} \end{bmatrix}$

and $\tilde{w}_{km} = [\bar{w}_{km} \bar{w}_{km}]$ respectively are the input membership functions and the output connecting weights.

The structure of the proposed NT2FC includes five layers, as shown in Fig. 1. The final output of the NT2FC is computed as the following steps:

**Step 1:** This step prepares the vector inputs $X = [x_1, x_2, \ldots, x_{n_i}]^T$ and $\dot{X} = [\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_{n_i}]^T$. Then these inputs will be directly transferred to the membership functions in the next step.

**Step 2:** The inputs $x_i$ and $\dot{x}_i$ in the previous step will be fed into the 3-D Gaussian membership functions in (8) and (9). A detailed illustration can be seen in Figure 2. The equation for computing the membership grades are given as:

$$
\mu_{ij} = \exp \left( -\frac{1}{2} \frac{(x_i - c_{ij})^2 + (\dot{x}_i - c_{ij})^2}{v_{ij}^2} \right) \tag{8}
$$

$$
\bar{\mu}_{ij} = \exp \left( -\frac{1}{2} \frac{(x_i - c_{ij})^2 + (\dot{x}_i - c_{ij})^2}{\bar{v}_{ij}^2} \right) \tag{9}
$$
where $\bar{\mu}_{ij}$ and $\bar{v}_{ij}$ respectively are the upper membership grades and upper variance of the GMFs; $\mu_{ij}$ and $v_{ij}$ respectively are the lower membership grades and lower variance of the GMFs; $c_{ij}$ is the mean of the GMFs.

**Step 3:** In this step, the product operation is applied to obtain the upper and lower fuzzy firing strengths as follows:

$$\bar{f}_k = \prod_{i=1}^{m} \bar{\mu}_{ij} \quad \text{and} \quad f_k = \prod_{i=1}^{m} \mu_{ij}$$

where $\bar{f}_k$ and $f_k$ respectively are the upper and lower fuzzy firing strengths.

**Step 4:** In this step, the centre-of-sums type-reduction is used to obtain the pre-output layer $o^l_m$ and $o^r_m$.

$$o^l_m = \frac{\sum_{k=1}^{n_k} (\bar{f}_k \bar{w}_{km})}{\sum_{k=1}^{n_k} (\bar{f}_k)}$$

$$o^r_m = \frac{\sum_{k=1}^{n_k} (\bar{f}_k \bar{w}_{km})}{\sum_{k=1}^{n_k} (\bar{f}_k)}$$

The output connecting weights $[w_{km} \bar{w}_{km}]$ using in this layer can be expressed as follows:

$$\begin{bmatrix} w_1^1 & w_1^2 & \cdots & w_1^{n_m} \\ w_2^1 & w_2^2 & \cdots & w_2^{n_m} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{w}_{k1}^1 & \bar{w}_{k1}^2 & \cdots & \bar{w}_{k1}^{n_m} \\ \bar{w}_{k2}^1 & \bar{w}_{k2}^2 & \cdots & \bar{w}_{k2}^{n_m} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{w}_{kn_k}^1 & \bar{w}_{kn_k}^2 & \cdots & \bar{w}_{kn_k}^{n_m} \end{bmatrix} \begin{bmatrix} \bar{w}_{11} & \bar{w}_{12} & \cdots & \bar{w}_{1n_m} \\ \bar{w}_{21} & \bar{w}_{22} & \cdots & \bar{w}_{2n_m} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{w}_{k1} & \bar{w}_{k2} & \cdots & \bar{w}_{kn_k} \\ \bar{w}_{k1} & \bar{w}_{k2} & \cdots & \bar{w}_{kn_k} \end{bmatrix}$$

**Step 5:** In this step, the defuzzification operator using the average of the left and right pre-output layer values

$$o_m = \frac{\sum_{k=1}^{n_k} (\bar{f}_k \bar{w}_{km})}{\sum_{k=1}^{n_k} (\bar{f}_k)}$$
where \( o'_m \) and \( o''_m \) respectively are the left and right pre-output nodes; \( \hat{u}_{NT2FC}^m \) is the control signal of \( m^{th} \) output using the proposed NT2FC controller.

The control vectors using the proposed NT2FC can be denoted as follows:

\[
\hat{u}_{NT2FC} = \left[ \hat{u}_{NT2FC}^1(t), \hat{u}_{NT2FC}^2(t), \ldots, \hat{u}_{NT2FC}^n(t) \right] \in \mathbb{R}^n
\]

**B. THE PARAMETER LEARNING FOR NT2FC**

The control vector \( \hat{u}_{NT2FC} \) in (16) is used to optimize the ideal synchronizer control vector \( \hat{u}^*(t) \) in (6). The Lyapunov function is defined as \( E = \frac{1}{2} (e_m(t))^2 \), where \( m \) can be replaced by \( x, y, z, v, \) and \( w \). Then, using the gradient descent approach and the chain rule, the adaptation laws of \( \hat{u}_{NT2FC} \) are derived as follows:

\[
\hat{\omega}_{km}(t + 1) = \hat{\omega}_{km}(t) - \hat{\eta}_w \frac{\partial E(t)}{\partial \hat{\omega}_{km}}
\]
FIGURE 9. The state trajectories of the hyperchaotic synchronization systems in 3-D using NT2FC synchronizer (a) $x - y - z$ orbits, (b) $x - y - v$ orbits, (c) $x - y - w$ orbits, (d) $z - v - w$ orbits.

\[
\begin{align*}
\dot{w}_{km}(t) &= \frac{\partial E(t)}{\partial e_m(t)} \cdot \partial m_2(t) \cdot \partial \omega_{MIT}^{2FC} \cdot \partial \omega_m \cdot \partial \omega_{km} \\
\dot{w}_{km}(t+1) &= \dot{w}_{km}(t) - \frac{1}{2} \dot{w}_{km}(t) \cdot m_2(t) \cdot \sum_{k=1}^{n_k} f_k \\
\dot{v}_{ij}(t+1) &= \dot{v}_{ij}(t) - \frac{\partial E(t)}{\partial \hat{v}_{ij}} \\
\dot{v}_{ij}(t) &= \frac{\partial E(t)}{\partial e_m(t)} \cdot \partial m_2(t) \cdot \partial \omega_{MIT}^{2FC} \cdot \partial \omega_m \cdot \partial \omega_{ij} \\
\dot{v}_{ij}(t+1) &= \dot{v}_{ij}(t) - \frac{1}{2} \dot{v}_{ij}(t) \cdot m_2(t) \cdot \sum_{k=1}^{n_k} f_k \\
\end{align*}
\]
\[ \dot{\hat{\eta}}_w = \dot{\hat{\eta}}_c = \dot{\hat{\eta}}_v = 0 \]

**Proof:**

\[ V(t) = E = \frac{1}{2} \left\{ (e_m^w(t))^2 + (e_m^c(t))^2 + (e_m^v(t))^2 \right\} \]
Thus,
\[
\Delta V(t) = V(t + 1) - V(t) = \frac{1}{2} \left[ \left( e^\eta_m(t + 1) \right)^2 - \left( e^\eta_m(t) \right)^2 \right] + \left[ \left( e^c_m(t + 1) \right)^2 - \left( e^c_m(t) \right)^2 \right] + \left[ \left( e^v_m(t + 1) \right)^2 - \left( e^v_m(t) \right)^2 \right]
\]
(23)

where
\[
e^\eta_m(t + 1) = e^\eta_m(t) + \Delta e^\eta_m(t) \doteqdot e_m(t) + \left[ \frac{\partial e_m(t)}{\partial \hat{w}_{km}} \right] \Delta \hat{w}_{km}
\]
(24)

\[
e^c_m(t + 1) = e^c_m(t) + \Delta e^c_m(t) \doteqdot e_m(t) + \left[ \frac{\partial e_m(t)}{\partial \hat{c}_{ij}} \right] \Delta \hat{c}_{ij}
\]
(25)

\[
e^v_m(t + 1) = e^v_m(t) + \Delta e^v_m(t) \doteqdot e_m(t) + \left[ \frac{\partial e_m(t)}{\partial \hat{v}_{ij}} \right] \Delta \hat{v}_{ij}
\]
(26)

From (17), (19), (20)
\[
\frac{\partial e_m(t)}{\partial \hat{w}_{km}} = e_m(t)m_2(t)\sum_{k=1}^{\tilde{f}_k} \frac{\tilde{f}_k}{m_k} = \varphi
\]
(27)
FIGURE 12. The synchronization error dynamics. (a) $e_x$, (b) $e_y$, (c) $e_z$, (d) $e_v$, and (e) $e_w$.

Rewrite (24)-(26) using (27)-(29) and (17) (19) (20), obtains

\[
\begin{align*}
\frac{\partial e_m(t)}{\partial c_{km}} &= e_m(t)m_2(t) \\
&\times \left( \sum_{k=1}^{n_k} (\tilde{f}_k) \right) \left( \left( \hat{w}_{km} - \sigma_m \right) + \left( \hat{x}_i - c_{ij} \right) \right) \\
&+ \frac{\tilde{f}_k}{\bar{v}_j} \left( \left( \hat{w}_{km} - \sigma_m \right) + \left( \hat{x}_i - c_{ij} \right) \right) \\
&= C
\end{align*}
\]

(28)

\[
\begin{align*}
\frac{\partial e_m(t)}{\partial v_{km}} &= e_m(t)m_2(t) \\
&\times \left( \sum_{k=1}^{n_k} (\tilde{f}_k) \right) \left( \left( \hat{w}_{km} - \sigma_m \right) + \left( \hat{x}_i - c_{ij} \right) \right) \\
&+ \frac{\tilde{f}_k}{\bar{v}_j} \left( \left( \hat{w}_{km} - \sigma_m \right) + \left( \hat{x}_i - c_{ij} \right) \right) \\
&= V
\end{align*}
\]

(29)

FIGURE 13. The learning rates online adjustment using MJA.

Rewrite (24)-(26) using (27)-(29) and (17) (19) (20), obtains

\[
e_m(t+1) = e_m(t) - \phi (\hat{\eta}_w e_m(t) \phi) \tag{30}
\]

\[
e_m(t+1) = e_m(t) - C (\hat{\eta}_c e_m(t) C) \tag{31}
\]
FIGURE 14. The state trajectories of the hyperchaotic synchronization systems in 3-D using NT2FC synchronizer (a) $x - y - z$ orbits, (b) $x - y - v$ orbits, (c) $x - y - w$ orbits, (d) $z - v - w$ orbits.

\[
e^m(t + 1) = e_m(t) - V \left( \hat{\eta}_w e_m(t) V \right) \tag{32}
\]

Rewrite (23) using (30)-(32), obtains

\[
\Delta V(t) = \frac{1}{2} \left( e_m(t) \right)^2 \left\{ \left( 1 - \hat{\eta}_w \varphi^2 \right)^2 - 1 \right\}
+ \left\{ \left( 1 - \hat{\eta}_c C^2 \right)^2 - 1 \right\} + \left\{ \left( 1 - \hat{\eta}_v V^2 \right)^2 - 1 \right\}
\]

\[
= \frac{1}{2} \hat{\eta}_w \left( e_m(t) \right)^2 \left\{ \varphi^2 \left( \hat{\eta}_w \varphi^2 - 2 \right) + C^2 \left( \hat{\eta}_c C^2 - 2 \right) + V^2 \left( \hat{\eta}_v V^2 - 2 \right) \right\} \tag{33}
\]

From (33), if $\hat{\eta}_w$ is chosen such that $0 < \hat{\eta}_w < \frac{2}{\varphi^2}$, $0 < \hat{\eta}_c < \frac{2}{\varphi^3}$, and $0 < \hat{\eta}_v < \frac{2}{V^2}$, then (33) is negative. Then, the stability of the synchronization system is ensured through the Lyapunov analysis.

C. LEARNING RATE OPTIMIZATION ALGORITHM

In order to optimize the learning rates for the adaptation law of the proposed network, the modified Jaya algorithm is designed as follow:

\textbf{Step 1:} Initialize $n_q$ set learning rates $\hat{\eta}_w$, $\hat{\eta}_c$, $\hat{\eta}_v$.

\textbf{Step 2:} For $i = 1$ to $n_q$; Run the NT2FC synchronization system with $i$-th learning rate set $\hat{\eta}_w^i$, $\hat{\eta}_c^i$, $\hat{\eta}_v^i$.

\textbf{Step 3:} Calculate the fitness function based on the synchronization errors. The root mean square error (RMSE) equation is defined as follows:

\[
f_e(t) = \sqrt{\frac{1}{n_q} \left( e_x(t) \right)^2 + \left( e_y(t) \right)^2 + \left( e_z(t) \right)^2 + \left( e_w(t) \right)^2 + \left( e_v(t) \right)^2}
\]

where $n_q$ is the number of samples.

\textbf{Step 4:} Choose the best and worst learning rate sets based on the lowest and highest fitness function, respectively. Obtain $\eta_w^{\text{best}}$, $\eta_c^{\text{best}}$, $\eta_v^{\text{best}}$ and $\eta_w^{\text{worst}}$, $\eta_c^{\text{worst}}$, $\eta_v^{\text{worst}}$. 
Step 5: For \( i = 1 \) to \( n_H \); Update the learning rate set \( \hat{\eta}_w, \hat{\eta}_c, \hat{\eta}_v \) by:

\[
\eta_{i\Sigma}^j(t + 1) = \eta_{i\Sigma}^j(t) + rd_1 \left[ \eta_{i\Sigma}^{\text{best}(rd)}(t) - \eta_{i\Sigma}^j(t) \right] - rd_2 \left[ \eta_{i\Sigma}^{\text{worst}(rd)}(t) - \eta_{i\Sigma}^j(t) \right]
\]  

(35)

where \( \Sigma \) can be replaced by \( w, c, \) or \( v \); \( rd \in [1, 3], rd_1 \in [0, 1] \) and \( rd_2 \in [0, 1] \) are the random numbers; \( \eta_{i\Sigma}^{\text{best}(rd)} \) and \( \eta_{i\Sigma}^{\text{worst}(rd)} \) respectively are the learning rates with \( rd \) lowest fitness function and highest fitness function.

Step 6: Run the NT2FC synchronization system with the updated learning rate sets. If new solutions in (35) are better than the old solutions, then accept the new \( i \)-th learning rate set \( \hat{\eta}_w, \hat{\eta}_c, \hat{\eta}_v \). Otherwise, keep the previous \( i \)-th learning rate set.

Step 7: Return to step 2 and repeat until the synchronization process is finished.

**FIGURE 15.** The state trajectories of master-slave hyperchaotic systems in time domain using the NT2FC. (a) \( x_1, x_2 \), (b) \( y_1, y_2 \), (c) \( z_1, z_2 \), (d) \( v_1, v_2 \) and (e) \( w_1, w_2 \).
Therefore, the NT2FC synchronizer can obtain suitable learning rates for the adaptation law by applying the proposed modified Jaya algorithm.

IV. ILLUSTRATIVE EXAMPLES
In this section, two examples of hyperchaotic synchronization are given to verify the validity of the proposed synchronization method. The synchronization scheme using the NT2FC synchronizer is demonstrated in Fig. 3. The parameters for the NT2FC synchronizer are set as $n_i = 5$, $n_j = 3$, $n_\eta = 20$; the sampling time is 0.001s; the initial parameters of hyperchaotic systems are set as $[x_1, y_1, z_1, v_1, w_1] = [5, 5, 5, 5]^T$ and $[x_2, y_2, z_2, v_2, w_2] = [0, 0, 0, 0]^T$, respectively. The external disturbances and system uncertainties are chosen as 

$$[ho_x(t), \rho_y(t), \rho_z(t), \rho_v(t), \rho_w(t)] = 0.2 [\cos \pi t, \cos \pi t, \cos \pi t, \cos \pi t]^T$$

and 

$$[\Delta f(x_2), \Delta f(y_2), \Delta f(z_2), \Delta f(v_2), \Delta f(w_2)] = rd(.01)[x_2, y_2, z_2, v_2, w_2]^T,$$

respectively.
Case 1:
In this case, as same with [48], the parameters of the 5-D hyperchaotic systems are set by \( a = 10, b = 8/3, c = 28, h_1 = -2, h_2 = -0.09, \) and \( h_3 = 8 \). The state trajectories of the hyperchaotic synchronization systems in 3-D using the NT2FC are displayed in Fig. 4. The state trajectories of master-slave hyperchaotic systems in time domain using the NT2FC are shown in Fig. 5. The synchronization control signals and the synchronization error dynamics are respectively given in Fig. 6 and Fig. 7. The learning rates adjustment using the MJA are shown in Fig. 8.

Case 2:
In this example, as same with [48], the parameters of the 5-D hyperchaotic systems are set by \( a = 10, b = 8/3, c = 28, h_1 = -1.155, h_2 = -0.12, \) and \( h_3 = 11.3 \). The state trajectories of the hyperchaotic synchronization systems in 3-D are displayed in Fig. 9. The state trajectories of master-slave hyperchaotic systems are shown in Fig. 10. The synchronization control signals and the synchronization error dynamics are respectively given in Fig. 11 and Fig. 12. The learning rates adjustment using the MJA are shown in Fig. 13.

Case 3:
In this example, the parameters of the 5-D hyperchaotic systems are chosen as same as case 2. In order to show the efficiency of the proposed method, the initial parameters of master and slave hyperchaotic systems are set different with the two cases above as: \( [x_1, y_1, z_1, v_1, w_1] = [4, 5, 2, 10, 36]^T \) and \( [x_2, y_2, z_2, v_2, w_2] = [2, -1, 7, 3, 15]^T \), respectively.
In this study, both online and offline MJA modes are used. First, the system will be run in offline mode 1000 times (each time 20 seconds) to calibrate the fitness function. After running the system with MJA offline mode, the best optimal learning rates will be selected to initialize the value for the MJA online mode (each time 0.1 seconds), the online MJA can learn and adapt quickly for real-time application. The fast adaptation and learning are due to the real-time feedback system.

Remark 5: In this study, the Lyapunov approach is used to find the learning rate ranges such that the system is stable. Besides, the MJA algorithm is also used to achieve optimal learning rates. Furthermore, the MJA algorithm can achieve optimal performance faster when the initial values are selected from that learning rate range.

V. CONCLUSION

In this paper, an NT2FC synchronizer is designed to synchronize a class of 5-D hyperchaotic systems. In the design of NT2FC, the 3-D type-2 Gaussian membership functions are applied to increase the ability to deal with uncertainties and disturbances. The MJA is applied to optimize the learning rates of the designed adaptation laws. Finally, the superiority of proposed method is examined on two simulation results of synchronization of the hyperchaotic systems. In summary, the proposed NT2FC is an efficient controller for the synchronization problems. Applying new algorithms to improve the system performance and using them to control other nonlinear systems will be our future work.

REFERENCES

[1] T.-L. Le, T.-T. Huynh, and S. K. Hong, “A modified grey wolf optimizer for optimum parameters of multilayer type-2 asymmetric fuzzy controller,” IEEE Access, vol. 8, pp. 121611–121629, 2020.
[2] C.-M. Lin, M.-H. Lin, and R.-G. Yeh, “Synchronization of unified chaotic system via adaptive wavelet cerebellar model articulation controller,” Neural Comput. Appl., vol. 23, nos. 3–4, pp. 965–973, Sep. 2013.
[3] M. R. Mufti, H. Afzal, F.-U. Rehman, Q. R. Butt, and M. I. Qureshi, “Synchronization and antisynchronization between two non-identical Chua oscillators via sliding mode control,” IEEE Access, vol. 6, pp. 45270–45280, 2018.
[4] X. Zhang, D. Li, and X. Zhang, “Adaptive fuzzy impulsive synchronization of chaotic systems with random parameters,” Chaos, Solitons Fractals, vol. 104, pp. 77–83, Nov. 2017.
[5] S.-Y. Li and M. A. B. Hernández, “Robust synchronization of chaotic systems with novel fuzzy rule-based controllers,” Inf. Sci., vol. 481, pp. 604–615, May 2019.
[6] L. Zhou, F. Tan, and F. Yu, “A robust synchronization-based chaotic secure communication scheme with double-layered and multiple hybrid networks,” IEEE Syst. J., vol. 14, no. 2, pp. 2508–2519, Jun. 2019.
[7] T.-L. Le, T.-T. Huynh, V.-Q. Nguyen, C.-M. Lin, and S.-K. Hong, “Chaotic synchronization using a self-evolving recurrent interval type-2 Petri cerebellar model articulation controller,” Mathematics, vol. 8, no. 2, p. 219, Feb. 2020.
[8] K. Tian, H.-P. Ren, and C. Bai, “Synchronization of hyperchaos with time delay using impulse control,” IEEE Access, vol. 8, pp. 72570–72576, 2020.
[9] O. Mofid, M. Momeni, S. Mobayen, and A. Fekih, “A disturbance-observer-based sliding mode control for the robust synchronization of uncertain delayed chaotic systems: Application to data security,” IEEE Access, vol. 9, pp. 16546–16555, 2021.
[10] J. Dai, Y. Cao, L. Xiao, H. Tan, and L. Jia, “Design and analysis of a noise-suppression zeroing neural network approach for robust synchronization of chaotic systems,” Neurocomputing, vol. 426, pp. 299–308, Feb. 2021.

TABLE 1. Comparison results in RMSE.

|            | Computation time (s) | Case 1 | Case 2 | Case 3 |
|------------|----------------------|--------|--------|--------|
| CMAC [34]  | 0.0169               | 0.8529 | 0.8316 | 0.8506 |
| IT2FNN [36]| 0.0162               | 0.7751 | 0.7625 | 0.7942 |
| NT2FC      | 0.0226               | 0.6849 | 0.7373 | 0.7447 |

FIGURE 18. The learning rates online adjustment using MJA.

The state trajectories of the hyperchaotic synchronization systems in 3-D are displayed in Fig. 14. The state trajectories of master-slave hyperchaotic systems using the proposed NT2FC are shown in Fig. 15. The synchronization control signals and the synchronization error dynamics are respectively given in Fig. 16 and Fig. 17. The learning rates adjustment using the MJA are shown in Fig. 18.

Remark 1: For three cases, the proposed NT2FC synchronizer has shown the effectiveness of synchronizing the 5-D hyperchaotic systems even when considering the external disturbances and system uncertainties. The RMSE among our proposed synchronizer and the cerebellar model articulation controller (CMAC), the interval type-2 fuzzy neural network (IT2FNN), are given in Table 1. The proposed NT2FC synchronizer has paid a little longer computation time than the other methods due to the MJA’s timely processing. However, it does not disturb the synchronization performance, and the proposed NT2FC synchronizer can obtain a better result in all cases.

Remark 2: The synchronization in case 3 had shown the robustness of the proposed method even when the initial parameters of hyperchaotic systems are set differently from case 1 and case 2.

Remark 3: The design of NT2FC’s structure such as the number of member functions and the number of fuzzy rules has a great influence on the performance of the system. In this study, we use the trial and error method to achieve a suitable network structure. The application of self-structured algorithms to achieve the optimal structure will be the next research direction.

Remark 4: In this study, both online and offline MJA modes are used. First, the system will be run in offline mode 1000 times (each time 20 seconds) to calibrate the fitness function. After running the system with MJA offline mode, the best optimal learning rates will be selected to initialize the value for the MJA online mode (each time 0.1 seconds), the online MJA can learn and adapt quickly for real-time application. The fast adaptation and learning are due to the real-time feedback system.

The state trajectories of the hyperchaotic synchronization systems in 3-D are displayed in Fig. 14. The state trajectories of master-slave hyperchaotic systems using the proposed NT2FC are shown in Fig. 15. The synchronization control signals and the synchronization error dynamics are respectively given in Fig. 16 and Fig. 17. The learning rates adjustment using the MJA are shown in Fig. 18.

Remark 1: For three cases, the proposed NT2FC synchronizer has shown the effectiveness of synchronizing the 5-D hyperchaotic systems even when considering the external disturbances and system uncertainties. The RMSE among our proposed synchronizer and the cerebellar model articulation controller (CMAC), the interval type-2 fuzzy neural network (IT2FNN), are given in Table 1. The proposed NT2FC synchronizer has paid a little longer computation time than the other methods due to the MJA’s timely processing. However, it does not disturb the synchronization performance, and the proposed NT2FC synchronizer can obtain a better result in all cases.

Remark 2: The synchronization in case 3 had shown the robustness of the proposed method even when the initial parameters of hyperchaotic systems are set differently from case 1 and case 2.

Remark 3: The design of NT2FC’s structure such as the number of member functions and the number of fuzzy rules has a great influence on the performance of the system. In this study, we use the trial and error method to achieve a suitable network structure. The application of self-structured algorithms to achieve the optimal structure will be the next research direction.

Remark 4: In this study, both online and offline MJA modes are used. First, the system will be run in offline mode 1000 times (each time 20 seconds) to calibrate the fitness function. After running the system with MJA offline mode, the best optimal learning rates will be selected to initialize the value for the MJA online mode (each time 0.1 seconds), the online MJA can learn and adapt quickly for real-time application. The fast adaptation and learning are due to the real-time feedback system.

Remark 5: In this study, the Lyapunov approach is used to find the learning rate ranges such that the system is stable. Besides, the MJA algorithm is also used to achieve optimal learning rates. Furthermore, the MJA algorithm can achieve optimal performance faster when the initial values are selected from that learning rate range.

V. CONCLUSION

In this paper, an NT2FC synchronizer is designed to synchronize a class of 5-D hyperchaotic systems. In the design of NT2FC, the 3-D type-2 Gaussian membership functions are applied to increase the ability to deal with uncertainties and disturbances. The MJA is applied to optimize the learning rates of the designed adaptation laws. Finally, the superiority of proposed method is examined on two simulation results of synchronization of the hyperchaotic systems. In summary, the proposed NT2FC is an efficient controller for the synchronization problems. Applying new algorithms to improve the system performance and using them to control other nonlinear systems will be our future work.
A. Mohammadzadeh and S. Ghaemi, "Finite-time synchronization of chaotic coronary artery system with input time-varying delay," *Chaos, Solitons Fractals*, vol. 134, May 2020, Art. no. 109683.

C. Huang, L. Cai, and J. Cao, "Linear control for synchronization of a fractional-order time-delayed chaotic financial system," *Chaos, Solitons Fractals*, vol. 113, pp. 326–332, Aug. 2018.

S. Vaidyanathan, "Adaptive anti-synchronization of 5-D novel hyper-chaotic systems and SPICE implementation," in *IEEE Int. Conf. Comput. Intel. Comput. Res.*, Dec. 2014, pp. 1–7.

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

S. Vaidyanathan, A. Sambas, A. T. Azar, "A 5-D hyperchaotic dynamo system with multi-stability, its dynamical analysis, active backstepping control, and circuit simulation," in *Backstepping Control of Nonlinear Dynamical Systems*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 449–471.

S. Vaidyanathan, A. Sambas, A. T. Azar, K. Rana, and V. Kumar, "A new 5-D hyperchaotic four-wing system with multi-stability and hidden attractor, its backstepping control, and circuit simulation," in *Backstepping Control of Nonlinear Dynamical Systems*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 115–138.

M. Hussam, "New lightweight hybrid encryption algorithm for cloud computing (LMGHMA-128bit) by using new 5-D hyperchaotic system," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 10, pp. 2531–2540, 2021.

F. Yang, J. Mou, Y. Cao, and R. Chu, "An image encryption algorithm based on BP neural network and hyperchaotic system," *China Commun.*, vol. 17, no. 5, pp. 21–28, May 2020.

M. R. Faiighi, H. Delavari, and D. Baleanu, "Control of an uncertain fractional-order liu system via fuzzy fractional-order sliding mode control," *J. Vibrat. Control*, vol. 18, no. 9, pp. 1366–1374, Aug. 2012.

H. Delavari and M. Shokrieh, "Fuzzy modeling and synchronization of a new hyperchaotic complex system with uncertainties," *J. Appl. Comput. Mech.*, vol. 1, no. 3, pp. 134–144, 2015.

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.

X. Gu, P. P. Angelov, and Q. Shen, "Self-organizing fuzzy belief inference system for classification," *IEEE Trans. Fuzzy Syst.*, early access, May 30, 2022, doi: 10.1109/TFUZZ.2022.3179148.

S.-K. Oh, H.-J. Jang, and W. Pedrycz, "A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11217–11229, Sep. 2011.

O. Castillo, R. Martínez-Marroquín, P. Melin, F. Valdez, and I. Soria, "Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot," *Inf. Sci.*, vol. 192, pp. 19–38, Jun. 2012.

O. Castillo, L. Amador-Angulo, J. R. Castro, and M. Garcia-Valdez, "A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems," *Inf. Sci.*, vol. 354, pp. 257–274, Aug. 2016.

K. A. Naik and C. P. Gupta, "Performance comparison of type-1 and type-2 fuzzy logic systems," in *Proc. 4th Int. Conf. Signal Process., Comput. Control (ISPCC)*, Sep. 2017, pp. 72–76.

M. Montazeri-Gh and S. Yazdani, "Application of interval type-2 fuzzy logic systems to gas turbine fault diagnosis," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106703.

Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, Oct. 2000.