IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 18, 2020, accepted March 13, 2020, date of publication March 16, 2020, date of current version March 26, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2981186*

# Self-Organizing Interval Type-2 Fuzzy Asymmetric CMAC Design to Synchronize Chaotic Satellite Systems Using a Modified Grey Wolf Optimizer

**TIEN-LOC LE[1,2], (Member, IEEE), TUAN-TU HUYNH[2,3], (Member, IEEE), AND SUNG-KYUNG HONG[1]**

[1]Faculty of Mechanical and Aerospace, Sejong University, Seoul 143-747(05006), South Korea
[2]Department of Electrical, Electronics, and Mechanical Engineering, Lac Hong University, Bien Hoa 810000, Vietnam
[3]Department of Electrical Engineering, Yuan Ze University, Taoyuan 320, Taiwan

Corresponding author: Sung-Kyung Hong (skhong@sejong.ac.kr)

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2018-0-01423) supervised by the IITP (Institute for Information and Communications Technology Promotion).

**ABSTRACT** This study presents a self-organizing interval type-2 fuzzy asymmetric cerebellar model articulation controller (MSIT2FAC) design for synchronizing chaotic satellite systems that use a modified grey wolf optimizer. The proposed control system uses MSIT2FAC as the main controller (which mimics an ideal controller) and a robust compensation controller (which addresses the approximation error between the ideal controller and the main controller). The self-organizing algorithm is used to generate the first network layer. In subsequent iterations, it autonomously increases or decreases the number of network layers using the tracking error. The adaptive laws for adjusting the parameters for the fuzzy rule for the proposed system are derived using the gradient descent method. The optimal learning rates for the adaptive laws are achieved using a modified grey wolf optimizer. The Lyapunov stability analysis guarantees the stability of the proposed algorithm. Finally, the numerical simulation results illustrate the effectiveness of the proposed method.

**INDEX TERMS** Interval type-2 fuzzy neural network, self-organizing algorithm, cerebellar model articulation controller, asymmetric membership function, chaotic satellite.

## I. INTRODUCTION

A satellite system exhibits chaotic behavior under the influence of gravitational and geomagnetic fields and solar radiation pressure [1]. Synchronization with other satellites increases the accuracy of satellite missions. The synchronization of chaotic satellite systems requires a control input that forces a slave satellite to conform with a master satellite. A chaotic satellite is a nonlinear system with a specific sensitivity to the initial conditions, a fractal structure and which is non-periodic [2]. Recent studies have proposed a variety of techniques and methodologies to synchronize a chaotic satellite system, such as fuzzy control [3]–[5], predictive control [6], sliding mode control [7], control using a neural network [1], and a linear matrix inequality [8]. However, most of these methods are complex and have scope for improvement in terms of synchronization performance. This study proposes a self-organizing interval type-2 fuzzy cerebellar model articulation controller (CMAC) with an asymmetric membership function (AMF) and a modified grey wolf optimizer (MGWO) that allows enhanced synchronization of a chaotic satellite system.

In recent years, CMAC has been widely used in various fields [9]–[15]. CMAC is a type of neural network that is based on a mammalian cerebellum model (associative memory). Its advantages are that it learns quickly, uses simple computation and can be generalized [16]. Gaussian membership functions (GMF) are commonly used in traditional CMAC networks because parameters can be easily adjusted. However, similarly to type-1 fuzzy logic systems (T1FLSs), a CMAC with a type-1 GMF (T1GMF) cannot deal with system uncertainties [17]. In 1975, Zadeh proposed the concept of

The associate editor coordinating the review of this manuscript and approving it for publication was Yongping Pan.

type-2 fuzzy logic systems (T2FLSs), which cope well with system uncertainties [18]. Interval T2FLSs (IT2FLSs) were used by Liang and Mendel as a simplified method to compute both the input and antecedent operations for a T2FLS, in order to reduce the complexity of calculations for these logic systems [19]. In recent years, T1FLSs and IT2FLSs have been the subject of many studies [20]–[26]. In 2017, Pan *et al.* proposed an adaptive fuzzy proportional derivative controller with a stable H $\infty$ tracking guarantee [20]. In 2018, Gil *et al.* proposed a fuzzy rule interpolation to optimize traffic light cycles [21]. In 2019, Soto *et al.* derived a multiple-input multiple-output fuzzy aggregation model to predict multiple time series [22]. The results of many studies show that IT2FLSs perform better than T1FLSs [27]–[30]. IT2FLSs better handle uncertainties and allow a more general design with more degrees of freedom [29]. To achieve better performance, some studies [17], [32]–[36] used a type-2 GMF (T2GMF) in CMAC structures. Symmetric and fixed membership functions (MF) are commonly used to simplify the design of CMAC or fuzzy systems. Some studies in [37], [38] show that a symmetric MF adversely affects a system's accuracy. Hellendoorn and Thomas used an AMF for fuzzy systems [39]. A type-2 AMF (T2AMF) usually consists of two GMFs as the upper MF and two GMFs as the lower MF so it accommodates an MF with an uncertain mean and an uncertain width and its learning capability and flexibility are increased [37], [40].

The learning rate for an adaptive controller significantly affects the performance of the system. If the learning rate is too small, convergence is very slow and the system is easily trapped in a local minimum. If the learning rate is too large, the system oscillates, is unstable and does not converge [41]. This study proposes a MGWO that improves the random searching positions and remembers the best solution, so the grey wolf optimizer (GWO) has increased search capability. The proposed algorithm is then used to optimize the learning rate for the adaptive laws. The GWO algorithm is a meta-heuristic algorithm that is inspired by the grey wolf community hierarchy and hunting mechanisms [42]. Several studies have used a GWO to solve real-life problems [43]–[48]. In 2017, Rodríguez *et al.* introduced a fuzzy hierarchical operator for a GWO [43]. In 2018, Qais *et al.* proposed an optimum parameter for multiple proportional-integral controllers using a GWO [44]. In 2019, Faris *et al.* demonstrated an automatic selection of hidden neurons and weights in neural networks using a GWO [45]. There are several other optimization algorithms. One study [49] used an enhanced adaptive fuzzy control that features optimal convergence for the approximation error. An accelerated cuckoo optimization algorithm has been used to solve capacitated vehicle routing problems [50]. A nature-inspired optimization algorithm for the fuzzy controller is proposed [51]. It is critical that the network has an appropriate size when designing a fuzzy controller or a CMAC controller [52]. Many studies use a trial-and-error approach to address this problem. This study proposes a network that uses a self-organizing algorithm to autonomously construct a network structure, which increases performance. This study constructs a self-organizing interval type-2 fuzzy asymmetric cerebellar model articulation controller using a MGWO (MSIT2FAC) that allows better synchronization for a chaotic satellite system. The research comprises the following: (1) An interval type-2 fuzzy CMAC controller with AMF is developed; (2) The parameters for the proposed system are updated online using the gradient descent method; (3) The AMF increases the learning capability and flexibility of the proposed network; (4) The optimal learning rates for the adaptive laws are derived using the MGWO; (5) The network structure and the parameters for the antecedent for the fuzzy rule are determined using the self-organizing algorithm. The main contribution of this study is the design of a new adaptive controller that combines the advantages of an interval type-2 fuzzy network, an asymmetric membership function, a modified grey wolf optimizer, and a self-organizing algorithm. Unlike earlier studies [53], [54], this study proposes a control method, that allows the proposed control system to increase the network's learning capability and flexibility to better deal with the uncertainties and for which it is easy to design the initial network parameters.

The remainder of this paper is structured as follows. Section 2 presents the equations for problem formulation in chaotic satellite systems. Section 3 proposes the structure of the MSIT2FCA, the parameter learning algorithm, the self-organizing algorithm, and the MGWO. Section 4 presents the simulation results of the chaotic satellite synchronization. Finally, Section 5 presents the study's conclusions.

## II. PROBLEM FORMULATION

A master and slave chaotic satellite systems in [3] is expressed as:

Master system:

$$\begin{aligned}
\dot{x}_1(t) &= \sigma_x x_2(t)x_3(t) - \frac{1.2}{I_x}x_1(t) + \frac{\sqrt{6}}{2I_x}x_3(t)\\
\dot{x}_2(t) &= \sigma_y x_1(t)x_3(t) + \frac{0.35}{I_y}x_2(t)\\
\dot{x}_3(t) &= \sigma_z x_1(t)x_2(t) - \frac{\sqrt{6}}{I_z}x_1(t) - \frac{0.4}{I_z}x_3(t)
\end{aligned} \tag{1}$$

Slave system:

$$\begin{aligned}
\dot{y}_1(t) &= \sigma_x y_2(t)y_3(t) - \frac{1.2}{I_x}y_1(t)\\
&\quad + \frac{\sqrt{6}}{2I_x}y_3(t) + d_1(t) + \Delta f(y_1) + u_1(t)\\
\dot{y}_2(t) &= \sigma_y y_1(t)y_3(t) + \frac{0.35}{I_y}y_2(t)\\
&\quad + d_2(t) + \Delta f(y_2) + u_2(t)\\
\dot{y}_3(t) &= \sigma_z y_1(t)y_2(t) - \frac{\sqrt{6}}{I_z}y_1(t)\\
&\quad - \frac{0.4}{I_z}y_3(t) + d_3(t) + \Delta f(y_3) + u_3(t)
\end{aligned} \tag{2}$$

where $\boldsymbol{x}(t) = [x_1(t), x_2(t), x_3(t)]$ and $\boldsymbol{y}(t) = [y_1(t), y_2(t), y_3(t)]$, respectively denote the chaotic positions of the master and slave systems; $I_x$, $I_y$, and $I_z$ are the principal moments of inertia; $\sigma_x, \sigma_y, \sigma_z$, are the chaotic coefficients, which are written as $\sigma_x = \frac{I_y - I_z}{I_x}$, $\sigma_y = \frac{I_z - I_x}{I_y}$, $\sigma_z = \frac{I_x - I_y}{I_z}$; $\boldsymbol{d}(t) = [d_1(t), d_2(t), d_3(t)]$ and $\Delta\boldsymbol{f}(y) = [\Delta(y_1), \Delta y_2(t), \Delta y_3(t)]$ respectively denote the external disturbances and the system uncertainties; and $\boldsymbol{u}(t) = [u_1(t), u_2(t), u_3(t)]$ is the signal for the input control torque.

Equations (1) and (2) can be rewritten in vector form as:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{G}(x(t)) \tag{3}$$

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{G}(y(t)) + \boldsymbol{d}(t) + \Delta\boldsymbol{f}(y(t)) + \boldsymbol{u}(t) \tag{4}$$

The vector synchronization errors for the two chaotic systems are defined as $\boldsymbol{e}(t) = [e_1(t), e_2(t), e_3(t)]$ and are written as:

$$\begin{aligned} e_1(t) &= y_1(t) - x_1(t) \\ e_2(t) &= y_2(t) - x_2(t) \\ e_3(t) &= y_3(t) - x_3(t) \end{aligned} \tag{5}$$

Therefore, the error dynamics are written as:

$$\begin{aligned} \dot{e}_1(t) &= \sigma_x \left(y_2(t)y_3(t) - x_2(t)x_3(t)\right) - \frac{1.2}{I_x}e_1 \\ &\quad + \frac{\sqrt{6}}{2I_x}e_3 + d_1(t) + \Delta f(y_1) + u_1(t) \\ \dot{e}_2(t) &= \sigma_y \left(y_1(t)y_3(t) - x_1(t)x_3(t)\right) + \frac{0.35}{I_y}e_2(t) \\ &\quad + d_2(t) + \Delta f(y_2) + u_2(t) \\ \dot{e}_3(t) &= \sigma_z \left(y_1(t)y_2(t) - x_1(t)x_2(t)\right) - \frac{\sqrt{6}}{I_z}e_1(t) \\ &\quad - \frac{0.4}{I_z}e_3(t) + d_3(t) + \Delta f(y_3) + u_3(t) \end{aligned} \tag{6}$$

Equation (6) can be rewritten in vector form as:

$$\dot{\boldsymbol{e}}(t) = \boldsymbol{G}\boldsymbol{e}(t) + \boldsymbol{f}(t) + \boldsymbol{d}(t) + \Delta\boldsymbol{f}(y(t)) + \boldsymbol{u}(t) \tag{7}$$

where

$$\boldsymbol{G} = \begin{bmatrix} \dfrac{-1.2}{I_x} & 0 & \dfrac{\sqrt{6}}{2I_x} \\ 0 & \dfrac{0.35}{I_y} & 0 \\ \dfrac{-\sqrt{6}}{2I_z} & 0 & \dfrac{-0.4}{I_z} \end{bmatrix},$$

and

$$\boldsymbol{f}(t) = \begin{bmatrix} \sigma_x \left(y_2(t)y_3(t) - x_2(t)x_3(t)\right) \\ \sigma_y \left(y_1(t)y_3(t) - x_1(t)x_3(t)\right) \\ \sigma_z \left(y_1(t)y_2(t) - x_1(t)x_2(t)\right) \end{bmatrix}$$

From (7), the ideal controller is designed as:

$$\boldsymbol{u}^*(t) = -\boldsymbol{G}\boldsymbol{e}(t) - \boldsymbol{f}(t) - \boldsymbol{d}(t) - \Delta\boldsymbol{f}(y(t)) - \dot{\boldsymbol{e}}(t) \tag{8}$$

However, the external disturbances and the system uncertainties in (8) cannot be exactly known, so this study uses a MSIT2FCA controller (which mimics the ideal controller) to synchronize the master and slave chaotic satellite systems.

## III. STRUCTURE OF THE MSIT2FCA

### A. T2FLS CMAC WITH AMF

For a class of T2FLS, the $\lambda^{\text{th}}$ fuzzy inference rule is:

$$\begin{aligned} &\textit{Rule } \lambda: \ \textit{IF } I_1 \textit{ is } \tilde{\mu}_{1jk} \textit{ and } I_2 \textit{ is } \tilde{\mu}_{2jk}, \ldots, \textit{ and } I_{n_i} \textit{ is } \tilde{\mu}_{n_i jk} \\ &\quad \textit{Then } \tilde{w}_{jk} = \left[\underline{w}_{jk}\ \overline{w}_{jk}\right] \quad \textit{for } \lambda = 1, 2, \ldots, n_\lambda; \\ &\quad i = 1, 2, \ldots, n_i; \quad j = 1, 2, \ldots, n_j; \ k = 1, 2, \ldots, n_k \end{aligned} \tag{9}$$

where $\tilde{\mu}_{ijk}$ and $\tilde{w}_{jk}$ are the input and output MFs, respectively; $n_\lambda$ is the total number of rules; $n_i$, $n_j$, $n_k$ are the number of inputs, the number of layers, and the number of blocks in each layer, respectively; $\underline{w}_{ij}$ and $\overline{w}_{ij}$ denote the lower and upper weight for the fuzzy rules.

Figure 1 shows the structure of a MSIT2FCA, which has five spaces: an input space, a membership space, a receptive-field space, a weight memory space, and an output space. The basic functions and signal propagation for each space are represented as follows:

1) *Input Space:* In this space, the input vector $\boldsymbol{I} = \left[I_1, I_2, \ldots, I_{n_i}\right]^T \epsilon \Re^{n_i}$ is directly transferred to the next space without any computation.

2) *Association Memory Space*: In this multilayered space, each layer is accumulated as a block. Each block executes a T2AMF. The output from this space is given by the input signal $\boldsymbol{I}_1$ and the T2AMF as:

$$\overline{\mu}_{ijk} = \begin{cases} \exp\left\{\dfrac{-\left(I_i - \overline{m}_{ijk}^l\right)^2}{2\left(\overline{\sigma}_{ijk}^l\right)^2}\right\}, & I_i \le \overline{m}_{ijk}^l \\ 1, & \overline{m}_{ijk}^l \le I_i \le \overline{m}_{ijk}^r \\ \exp\left\{\dfrac{-\left(I_i - \overline{m}_{ijk}^r\right)^2}{2\left(\overline{\sigma}_{ijk}^r\right)^2}\right\}, & \overline{m}_{ijk}^r \le I_i \end{cases} \tag{10}$$

$$\overline{\mu}_{ijk} = \begin{cases} \underline{r}^*\exp\left\{\dfrac{-\left(I_i - \underline{m}_{ijk}^l\right)^2}{2\left(\underline{\sigma}_{ijk}^l\right)^2}\right\}, & I_i \le \underline{m}_{ijk}^l \\ \underline{r}, & \underline{m}_{ijk}^l \le I_i \le \underline{m}_{ijk}^r \\ \underline{r}^*\exp\left\{\dfrac{-\left(I_i - \underline{m}_{ijk}^r\right)^2}{2\left(\underline{\sigma}_{ijk}^r\right)^2}\right\}, & \underline{m}_{ijk}^r \le I_i \end{cases} \tag{11}$$

where $\overline{\mu}_{ijk}$ and $\underline{\mu}_{ijk}$ denote the upper and lower MFs; $\overline{m}_{ijk}^l$ and $\overline{m}_{ijk}^r$ a re the means of the two upper GMFs, $\overline{\sigma}_{ijk}^l$ and $\overline{\sigma}_{ijk}^r$ are the variances of the two upper GMFs, $\underline{m}_{ijk}^l$ and $\underline{m}_{ijk}^r$ are the means of the two lower GMFs, and $\underline{\sigma}_{ijk}^l$ and $\underline{\sigma}_{ijk}^r$ are the variances of the two upper GMFs. Fig. 2 shows the T2AMF, which uses four GMFs. The following constraints ensure a reasonable MF:

$$\begin{cases} \overline{m}_{ijk}^l \le \underline{m}_{ijk}^l \le \underline{m}_{ijk}^r \le \overline{m}_{ijk}^r \\ \underline{\sigma}_{ijk}^l \le \overline{\sigma}_{ijk}^l \le \underline{\sigma}_{ijk}^r \le \overline{\sigma}_{ijk}^r \\ 0.5 \le \underline{r} \le 1 \end{cases} \tag{12}$$

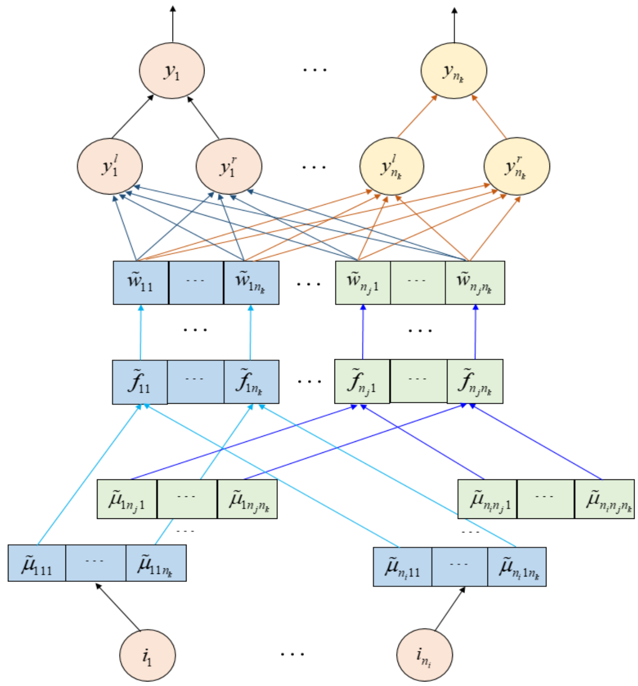

**FIGURE 1.** The structure of the MSIT2FCA control system.

3) *Receptive-Field Space*: Using a t-norm operation, the interval value for the multidimensional receptive field is described as

$$\underline{f}_{jk} = \prod_{i=1}^{n_i} \underline{\mu}_{ijk} \text{ and } \overline{f}_{jk} = \prod_{i=1}^{n_i} \overline{\mu}_{ijk} \tag{13}$$

4) *Weight Memory Space:* In this space, the rule consequence performs fuzzy operations. The adjustable connecting weight is denoted by $\tilde{\boldsymbol{w}}_{jk}$, which is defined as:

$$\tilde{\boldsymbol{w}}_{jk} = \left[\underline{\boldsymbol{w}}_{jk} \; \overline{\boldsymbol{w}}_{jk}\right] \tag{14}$$

where

$$\underline{\boldsymbol{w}}_{jk} = \left[\underline{w}_{11}, \dots \underline{w}_{1n_k}, \dots, \underline{w}_{n_j 1}, \dots, \underline{w}_{n_j n_k}\right] \in \Re^{n_j n_k}$$
$$\overline{\boldsymbol{w}}_{jk} = \left[\overline{w}_{11}, \dots \overline{w}_{1n_k}, \dots, \overline{w}_{n_j 1}, \dots, \overline{w}_{n_j n_k}\right] \in \Re^{n_j n_k}$$

5) *Output Layer:* This space uses the receptive-field space and the connecting weight memory space and functions as a de-fuzzifier:

$$\boldsymbol{u}^k_{MSIT2FCA} = \frac{y^l_k + y^r_k}{2} = \frac{1}{2}\left[\frac{\sum_{j=1}^{n_j} \underline{f}_{jk} \underline{w}_{jk}}{\sum_{j=1}^{n_j} \underline{f}_{jk}} + \frac{\sum_{j=1}^{n_j} \overline{f}_{jk} \overline{w}_{jk}}{\sum_{j=1}^{n_j} \overline{f}_{jk}}\right] \tag{15}$$

The control signal, $\boldsymbol{u}^k_{MSIT2FCA}$, is then used to mimic the ideal controller in (8) to achieve the desired control performance. The adaptive laws for updating controller parameters are detailed in the following section.

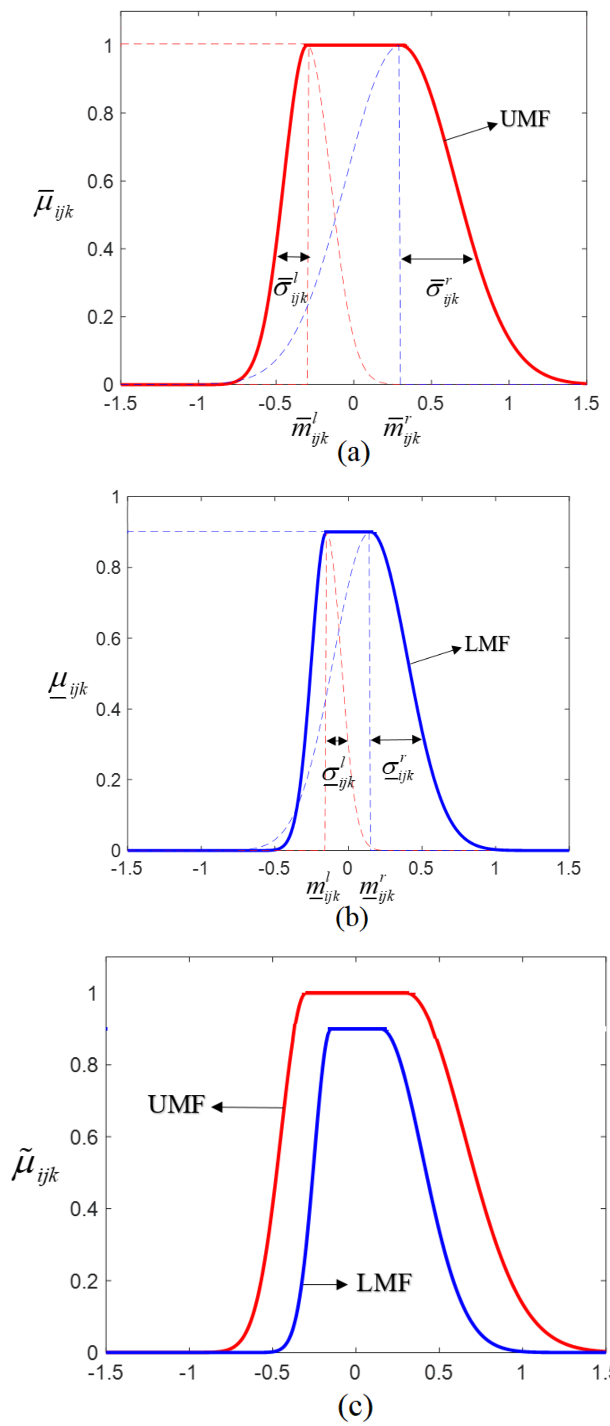


**FIGURE 2.** The AMF of the MSIT2FCA control system. (a) The upper MF, (b) the lower MF, and (c) the combined MF.

### *B. LEARNING THE PARAMETERS FOR THE MSIT2FCA*

The optimal controller $\boldsymbol{u}^*_{MSIT2FCA}$ in (15) is used to approximate the ideal controller in (8), so:

$$\boldsymbol{u}^*(t) = \boldsymbol{u}^*_{MSIT2FCA}(\underline{w}^*, \overline{w}^*, \underline{m}^{l*}, \overline{m}^{l*}, \underline{m}^{r*}, \overline{m}^{r*}, \underline{\sigma}^{l*}, \overline{\sigma}^{l*}, \underline{\sigma}^{r*}, \overline{\sigma}^{r*}, t) + \boldsymbol{\zeta}(t) \tag{16}$$

where $\underline{w}^*, \overline{w}^*, \underline{m}^{l*}, \overline{m}^{l*}, \underline{m}^{r*}, \overline{m}^{r*}, \underline{\sigma}^{l*}, \overline{\sigma}^{l*}, \underline{\sigma}^{r*}, \overline{\sigma}^{r*}$ are the optimal parameters for $\underline{w}, \overline{w}, \underline{m}^l, \overline{m}^l, \underline{m}^r, \overline{m}^r, \underline{\sigma}^l, \overline{\sigma}^l, \underline{\sigma}^r, \overline{\sigma}^r$, and the approximation error between the ideal controller and the optimal controller is denoted by $\boldsymbol{\zeta}(t)$.

The optimal parameters $\underline{w}^*, \overline{w}^*, \underline{m}^{l*}, \overline{m}^{l*}, \underline{m}^{r*}, \overline{m}^{r*}, \underline{\sigma}^{l*}, \overline{\sigma}^{l*}, \underline{\sigma}^{r*}, \overline{\sigma}^{r*}$ cannot be obtained, so an estimation controller $\hat{\boldsymbol{u}}_{MSIT2FCA}$ estimates (16) as

$$\hat{\boldsymbol{u}}\,(t) = \hat{\boldsymbol{u}}_{MSIT2FCA}(\hat{\underline{w}}, \hat{\overline{w}}, \hat{\underline{m}}^l, \hat{\overline{m}}^l, \hat{\underline{m}}^r, \hat{\overline{m}}^r, \hat{\underline{\sigma}}^l, \hat{\overline{\sigma}}^l, \hat{\underline{\sigma}}^r, \hat{\overline{\sigma}}^r, t) + \hat{\boldsymbol{u}}_{RB}\,(t) \tag{17}$$

where $\hat{\underline{w}}, \hat{\overline{w}}, \hat{\underline{m}}^l, \hat{\overline{m}}^l, \hat{\underline{m}}^r, \hat{\overline{m}}^r, \hat{\underline{\sigma}}^l, \hat{\overline{\sigma}}^l, \hat{\underline{\sigma}}^r, \hat{\overline{\sigma}}^r$ are the estimations of $\underline{w}^*, \overline{w}^*, \underline{m}^{l*}, \overline{m}^{l*}, \underline{m}^{r*}, \overline{m}^{r*}, \underline{\sigma}^{l*}, \overline{\sigma}^{l*}, \underline{\sigma}^{r*}, \overline{\sigma}^{r*}$ and $\hat{\boldsymbol{u}}_{RB}$ is the robust compensator controller, which is used to eliminate $\boldsymbol{\zeta}(t)$ in (16).

The robust compensator controller is defined as

$$\boldsymbol{u}_{RB}(t) = \hat{\boldsymbol{R}}(t)sgn(\boldsymbol{s}(t)) \tag{18}$$

The adaptive law for updating the uncertainty bound $\hat{\boldsymbol{R}}(t)$ is:

$$\dot{\hat{\boldsymbol{R}}}(t) = \eta_R\,|\boldsymbol{s}(t)| \tag{19}$$

where $\eta_R$ is the learning rate for updating $\hat{\boldsymbol{R}}(t)$; $\boldsymbol{s}(t)$ is the high-order sliding surface, which is defined as

$$\boldsymbol{s}(t) = \boldsymbol{e}^{(n-1)} + k_1\boldsymbol{e}^{(n-2)} \ldots + k_n \int_0^t \boldsymbol{e}\,(\tau)d\tau \tag{20}$$

Taking the derivative of (20), then

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{e}^{(n)} + \boldsymbol{K}^T\boldsymbol{e} \tag{21}$$

where $\boldsymbol{K} = [k_1, k_2, \ldots, k_n]$ is the feedback gain vector.

A Lyapunov function is defined as

$$V_1\,(\boldsymbol{s}\,(t)) = \frac{1}{2}\boldsymbol{s}^2\,(t) \tag{22}$$

Taking the derivative of (22) and using (7), (17) and (21), given:

$$\begin{aligned}
&\dot{V}_1(\boldsymbol{s}(t))\\
&= \boldsymbol{s}(t)\dot{\boldsymbol{s}}(t) = \boldsymbol{s}(t)\left[\boldsymbol{e}^{(n)} + \boldsymbol{K}^T\boldsymbol{e}\right]\\
&= \boldsymbol{s}(t)\Big[\boldsymbol{Ge}(t) + \boldsymbol{f}(t) + \boldsymbol{d}(t) + \Delta\boldsymbol{f}\,(y(t))\\
&\quad + \Big(\hat{\boldsymbol{u}}_{MSIT2FCA}(\hat{\underline{w}}, \hat{\overline{w}}, \hat{\underline{m}}^l, \hat{\overline{m}}^l, \hat{\underline{m}}^r, \hat{\overline{m}}^r, \hat{\underline{\sigma}}^l, \hat{\overline{\sigma}}^l, \hat{\underline{\sigma}}^r, \hat{\overline{\sigma}}^r, t)\\
&\quad +\ \hat{\boldsymbol{u}}_{RB}\,(t)\Big) + \boldsymbol{K}^T\boldsymbol{e}\Big]
\end{aligned} \tag{23}$$

The objective is to tune the value of $\hat{\underline{w}}, \hat{\overline{w}}, \hat{\underline{m}}^l, \hat{\overline{m}}^l, \hat{\underline{m}}^r, \hat{\overline{m}}^r, \hat{\underline{\sigma}}^l, \hat{\overline{\sigma}}^l, \hat{\underline{\sigma}}^r, \hat{\overline{\sigma}}^r$ such that $\dot{V}_1(\boldsymbol{s}(t))$ is minimized, so $\boldsymbol{s}(t)$ converges rapidly. Using the gradient descent method, the parameters for the MSIT2FCA control system are updated as:

$$\begin{aligned}
&\hat{\underline{w}}_{jk}(t+1)\\
&= \hat{\underline{w}}_{jk}(t) - \hat{\eta}_w \frac{\partial \dot{V}_1(t)}{\partial \hat{\underline{w}}_{jk}}\\
&= \hat{\underline{w}}_{jk}(t) - \hat{\eta}_w \frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MSIT2FCA}} \frac{\partial \hat{u}_{MSTT2FCA}}{\partial y_k^l} \frac{\partial y_k^l}{\partial \hat{\underline{w}}_{jk}}\\
&= \hat{\underline{w}}_{jk}(t) - \frac{1}{2}\hat{\eta}_w s(t) \frac{\underline{f}_{jk}}{\sum_{j=1}^{n_j} \underline{f}_{jk}}
\end{aligned} \tag{24}$$

$$\begin{aligned}
&\hat{\overline{w}}_{jk}(t+1)\\
&= \hat{\overline{w}}_{jk}(t) - \hat{\eta}_w \frac{\partial \dot{V}_1(t)}{\partial \hat{\overline{w}}_{jk}}\\
&= \hat{\overline{w}}_{jk}(t) - \hat{\eta}_w \frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MSTT2FCA}} \frac{\partial \hat{u}_{MST2FCA}}{\partial y_k^r} \frac{\partial y_k^r}{\partial \hat{\overline{w}}_{jk}}\\
&= \hat{\overline{w}}_{jk}(t) - \frac{1}{2}\hat{\eta}_w s(t) \frac{\overline{f}_{jk}}{\sum_{j=1}^{n_j} \overline{f}_{jk}}
\end{aligned} \tag{25}$$

$$\begin{aligned}
&\hat{\underline{m}}_{ijk}^l(t+1)\\
&= \hat{\underline{m}_{ijk}}(t) - \hat{\eta}_m \frac{\partial \dot{V}_1(t)}{\partial \hat{\underline{m}}_{ijk}^l}\\
&= \hat{\underline{m}}_{ijk}^l(t) - \hat{\eta}_m \left(\frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MSTT2FCA}} \frac{\partial \hat{u}_{MSIT2FCA}}{\partial \hat{y}_k^l} \frac{\partial \hat{y}_k^l}{\partial \underline{f}_{jk}} \frac{\partial \underline{f}_{jk}}{\partial \underline{\mu}_{ijk}} \frac{\partial \underline{\mu}_{ijk}^l}{\partial \hat{\underline{m}}_{ijk}^l}\right)\\
&= \hat{\underline{m}}_{ijk}^l(t) - \frac{1}{2}\hat{\eta}_m s(t) \left(\frac{\underline{w}_{jk} - \hat{y}_k^l}{\sum_{j=1}^{n_j} \underline{f}_{jk}}\right) \left(\frac{\underline{f}_{jk}}{\underline{\mu}_{ijk}}\right) \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}_{ijk}^l}
\end{aligned} \tag{26}$$

$$\begin{aligned}
&\hat{\underline{m}}_{ijk}^r(t+1)\\
&= \hat{\underline{m}_{ijk}}^r(t) - \hat{\eta}_m \frac{\partial \dot{V}_1(t)}{\partial \hat{\underline{m}}_{ijk}^r}\\
&= \hat{\underline{m}_{jk}^r}(t) - \hat{\eta}_m \left(\frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MST2FCA}} \frac{\partial \hat{u}_{MST2FCA}}{\partial \hat{y}_k^T} \frac{\partial \hat{y}_k'}{\partial \underline{f}_{jk}} \frac{\partial \underline{f}_{jk}}{\partial \underline{\mu}_{jk}} \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}_{ijk}}\right)\\
&= \hat{\underline{m}}_{ijk}^r(t) - \frac{1}{2}\hat{\eta}_m s(t) \left(\frac{\underline{w}_{jk} - \hat{y}_k^l}{\sum_{j=1}^{n_j} \underline{f}_{jk}}\right) \left(\frac{\underline{f}_{jk}}{\underline{\mu}_{ijk}}\right) \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}_{ijk}^r}
\end{aligned} \tag{27}$$

$$\begin{aligned}
&\hat{\underline{\sigma}}_{ijk}^j(t+1)\\
&= \hat{\underline{\sigma}}_{ijk}^j(t) - \hat{\eta}_\sigma \frac{\partial \dot{V}_1(t)}{\partial \hat{\underline{\sigma}}_{ijk}'}\\
&= \hat{\underline{\sigma}_{jk}^l}(t) - \hat{\eta}_\sigma \left(\frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MSTT2FCA}} \frac{\partial \hat{u}_{MSTT2FCA}}{\partial \hat{y}_k^l} \frac{\partial \hat{y}_k^l}{\partial \underline{f}_{jk}} \frac{\partial \underline{f}_{jk}}{\partial \underline{\mu}_{ijk}} \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}_{ijk}^l}\right)\\
&= \hat{\underline{\sigma}}_{ijk}^l(t) - \frac{1}{2}\hat{\eta}_\sigma s(t) \left(\frac{\underline{w}_{jk} - \hat{y}_k^l}{\sum_{j=1}^{n_j} \underline{f}_{jk}}\right) \left(\frac{\underline{f}_{jk}}{\underline{\mu}_{ijk}}\right) \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}_{ijk}^l}
\end{aligned} \tag{28}$$

$$\begin{aligned}
&\hat{\underline{\sigma}}_{ijk}^r(t+1)\\
&= \hat{\underline{\sigma}}_{ijk}^r(t) - \hat{\eta}_\sigma \frac{\partial \dot{V}_1(t)}{\partial \hat{\underline{\sigma}}_{ijk}^r}\\
&= \hat{\underline{\sigma}}_{ijk}^r(t) - \hat{\eta}_\sigma \left(\frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MsTT2FCA}} \frac{\partial \hat{u}_{MSTT2FCA}}{\partial \hat{y}_k^l} \frac{\partial \hat{y}_k^l}{\partial \underline{f}_{jk}} \frac{\partial \underline{f}_{jk}}{\partial \underline{\mu}_{ijk}} \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}_{ijk}^r}\right)\\
&= \hat{\underline{\sigma}}_{ijk}^r(t) - \frac{1}{2}\hat{\eta}_\sigma s(t) \left(\frac{\underline{w}_{jk} - \hat{y}_k^l}{\sum_{j=1}^{n_j} \underline{f}_{jk}}\right) \left(\frac{\underline{f}_{jk}}{\underline{\mu}_{ijk}}\right) \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}_{ijk}^r}
\end{aligned} \tag{29}$$

$$\begin{aligned}
&\hat{\overline{m}}_{ijk}^l(t+1)\\
&= \hat{\overline{m}}_{ijk}^l(t) - \hat{\eta}_m \frac{\partial \dot{V}_1(t)}{\partial \hat{\overline{m}}_{ijk}^l}
\end{aligned}$$

$$= \hat{\bar{m}}^l_{ijk}(t) - \hat{\eta}_m \left( \frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MSTT2FCA}} \frac{\partial \hat{u}_{MSTT2FCA}}{\partial \hat{y}^r_k} \frac{\partial \bar{f}_{jk}}{\partial \bar{f}_{jk}} \frac{\partial \bar{\mu}_{ijk}}{\partial \bar{\mu}_{ijk}} \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{m}}^l_{ijk}} \right)$$

$$= \hat{\bar{m}}^l_{ijk}(t) - \frac{1}{2}\hat{\eta}_m s(t) \left( \frac{\bar{w}_{jk} - \hat{y}^r_k}{\sum_{j=1}^{n_j} \bar{f}_{jk}} \right) \left( \frac{\bar{f}_{jk}}{\bar{\mu}_{ijk}} \right) \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{m}}^l_{ijk}} \tag{30}$$

$$\hat{\bar{m}}^r_{ijk}(t+1)$$

$$= \hat{\underline{m}}^r_{ijk}(t) - \hat{\eta}_m \frac{\partial \dot{V}_1(t)}{\partial \hat{\underline{m}}^r_{ijk}}$$

$$= \hat{\bar{m}}^r_{ijk}(t) - \hat{\eta}_m \left( \frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MSTT2FCA}} \frac{\partial \hat{u}_{MSTT2FCA}}{\partial \hat{y}^r_k} \frac{\partial \hat{y}^r_k}{\partial \bar{f}_{jk}} \frac{\partial \bar{f}_{jk}}{\partial \bar{\mu}_{ijk}} \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{m}}^r_{ijk}} \right)$$

$$= \hat{\bar{m}}^r_{ijk}(t) - \frac{1}{2}\hat{\eta}_m s(t) \left( \frac{\bar{w}_{jk} - \hat{y}^r_k}{\sum_{j=1}^{n_j} \bar{f}_{jk}} \right) \left( \frac{\bar{f}_{jk}}{\bar{\mu}_{ijk}} \right) \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{m}}^r_{ijk}} \tag{31}$$

$$\hat{\bar{\sigma}}^J_{ijk}(t+1)$$

$$= \hat{\bar{\sigma}}^J_{ijk}(t) - \hat{\eta}_\sigma \frac{\partial \dot{V}_1(t)}{\partial \hat{\bar{\sigma}}^J_{ijk}}$$

$$= \hat{\bar{\sigma}}^l_{ijk}(t) - \hat{\eta}_\sigma \left( \frac{\partial \dot{V}_1(t)}{\partial \hat{u}_{MST2FCA}} \frac{\partial \hat{u}_{MST2FCA}}{\partial \hat{y}^r_k} \frac{\partial \hat{y}^r_k}{\partial \bar{f}_{jk}} \frac{\partial \bar{f}_{jk}}{\partial \bar{\mu}_{jk}} \frac{\partial \bar{\mu}_{jk}}{\partial \hat{\bar{\sigma}}^J_{ijk}} \right)$$

$$= \hat{\bar{\sigma}}^l_{ijk}(t) - \frac{1}{2}\hat{\eta}_\sigma s(t) \left( \frac{\bar{w}_{jk} - \hat{y}^r_k}{\sum_{j=1}^{n_j} \bar{f}_{jk}} \right) \left( \frac{\bar{f}_{jk}}{\bar{\mu}_{ijk}} \right) \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{\sigma}}^l_{ijk}} \tag{32}$$

$$\hat{\bar{\sigma}}^r_{ijk}(t+1)$$

$$= \hat{\bar{\sigma}}^r_{ijk}(t) - \hat{\eta}_\sigma \frac{\partial \dot{V}_1(t)}{\partial \hat{\bar{\sigma}}^r_{ijk}}$$

$$= \hat{\bar{\sigma}}^r_{ijk}(t) - \frac{1}{2}\hat{\eta}_\sigma s(t) \left( \frac{\bar{w}_{jk} - \hat{y}^r_k}{\sum_{j=1}^{n_j} \bar{f}_{jk}} \right) \left( \frac{\bar{f}_{jk}}{\bar{\mu}_{ijk}} \right) \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{\sigma}}^r_{ijk}} \tag{33}$$

where $\hat{\eta}_w, \hat{\eta}_m, \hat{\eta}_\sigma$ are the positive learning rates.

In (26)–(33), terms $\frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}^l_{ijk}}, \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}^r_{ijk}}, \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}^l_{ijk}}, \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}^r_{ijk}}$ and $\frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{m}}^l_{ijk}}, \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{m}}^r_{ijk}}, \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{\sigma}}^l_{ijk}}, \frac{\partial \bar{\mu}_{ijk}}{\partial \hat{\bar{\sigma}}^r_{ijk}}$ are calculated depending on the input region as presented below.

**For updating the lower MFs:**

Region (I): $I_i \le \underline{m}^l_{ijk}$

$$\frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}^l_{ijk}} = \underline{\mu}_{ijk} \frac{\left(I_i - \underline{m}^l_{ijk}\right)}{\left(\underline{\sigma}^l_{ijk}\right)^2}, \quad \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}^l_{ijk}} = \underline{\mu}_{ijk} \frac{\left(I_i - \underline{m}^l_{ijk}\right)^2}{\left(\underline{\sigma}^l_{ijk}\right)^3},$$
$$\frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}^r_{ijk}} = 0, \quad \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}^r_{ijk}} = 0 \tag{34}$$

Region (II): $\underline{m}^l_{ijk} \le I_i \le \underline{m}^r_{ijk}$

$$\frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}^l_{ijk}} = 0, \quad \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}^l_{ijk}} = 0, \quad \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}^r_{ijk}} = 0, \quad \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}^r_{ijk}} = 0 \tag{35}$$

Region (III): $\underline{m}^r_{ijk} \le I_i$

$$\frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}^l_{ijk}} = 0, \quad \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}^l_{ijk}} = 0, \quad \frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{m}}^r_{ijk}} = \underline{\mu}_{ijk} \frac{\left(I_i - \underline{m}^r_{ijk}\right)}{\left(\underline{\sigma}^r_{ijk}\right)^2},$$
$$\frac{\partial \underline{\mu}_{ijk}}{\partial \hat{\underline{\sigma}}^r_{ijk}} = \underline{\mu}_{ijk} \frac{\left(I_i - \underline{m}^r_{ijk}\right)^2}{\left(\underline{\sigma}^r_{ijk}\right)^3} \tag{36}$$

**For updating the upper MFs:**

Region (I): $I_i \le \bar{m}^l_{ijk}$

$$\frac{\partial \bar{\mu}_{ijk}}{\partial \bar{m}^l_{ijk}} = \bar{\mu}_{ijk} \frac{\left(I_i - \bar{m}^l_{ijk}\right)}{\left(\bar{\sigma}^l_{ijk}\right)^2}, \quad \frac{\partial \bar{\mu}_{ijk}}{\partial \bar{\sigma}^l_{ijk}} = \bar{\mu}_{ijk} \frac{\left(I_i - \bar{m}^l_{ijk}\right)^2}{\left(\bar{\sigma}^l_{ijk}\right)^3},$$
$$\frac{\bar{\mu}_{ijk}}{\partial \bar{m}^r_{ijk}} = 0, \quad \frac{\partial \bar{\mu}_{ijk}}{\partial \bar{\sigma}^r_{ijk}} = 0 \tag{37}$$

Region (II): $\bar{m}^l_{ijk} \le I_i \le \bar{m}^r_{ijk}$

$$\frac{\partial \bar{\mu}_{ijk}}{\partial \bar{m}^l_{ijk}} = 0, \quad \frac{\partial \bar{\mu}_{ijk}}{\partial \bar{\sigma}^l_{ijk}} = 0, \quad \frac{\bar{\mu}_{ijk}}{\partial \bar{m}^r_{ijk}} = 0, \quad \frac{\partial \bar{\mu}_{ijk}}{\partial \bar{\sigma}^r_{ijk}} = 0 \tag{38}$$

Region (III): $\bar{m}^r_{ijk} \le I_i$

$$\frac{\partial \bar{\mu}_{ijk}}{\partial \bar{m}^l_{ijk}} = 0, \quad \frac{\partial \bar{\mu}_{ijk}}{\partial \bar{\sigma}^l_{ijk}} = 0, \quad \frac{\bar{\mu}_{ijk}}{\partial \bar{m}^r_{ijk}} = \bar{\mu}_{ijk} \frac{\left(I_i - \bar{m}^r_{ijk}\right)}{\left(\bar{\sigma}^r_{ijk}\right)^2},$$
$$\frac{\partial \bar{\mu}_{ijk}}{\partial \bar{\sigma}^r_{ijk}} = \bar{\mu}_{ijk} \frac{\left(I_i - \bar{m}^r_{ijk}\right)^2}{\left(\bar{\sigma}^r_{ijk}\right)^3} \tag{39}$$

By utilizing the adaptive laws given in (24)–(33), the estimation parameters for $\hat{\boldsymbol{u}}_{MSIT2FCA}$ can be obtained.

**Proving the convergence:**

$$\text{Defined as } \boldsymbol{\zeta}_\varphi(t) = \frac{\partial \hat{u}_{MSIT2FCA}}{\partial \boldsymbol{\varphi}} \tag{40}$$

where $\boldsymbol{\varphi} = \hat{\underline{\boldsymbol{w}}}, \hat{\bar{\boldsymbol{w}}}, \hat{\underline{\boldsymbol{m}}}^l, \hat{\bar{\boldsymbol{m}}}^l, \hat{\underline{\boldsymbol{m}}}^r, \hat{\bar{\boldsymbol{m}}}^r, \hat{\underline{\boldsymbol{\sigma}}}^l, \hat{\bar{\boldsymbol{\sigma}}}^l, \hat{\underline{\boldsymbol{\sigma}}}^r, \hat{\bar{\boldsymbol{\sigma}}}^r$

Consider the derivative of the Lyapunov function in (22) using the gradient descent method, which is obtained by:

$$\begin{aligned} \dot{V}(s(t+1)) &= \dot{V}(s(t)) + \Delta\dot{V}(s(t)) \\ &\cong \dot{V}(s(t)) + \left[\frac{\partial \dot{V}(s(t))}{\partial \varphi}\right]^T \Delta\varphi \end{aligned} \tag{41}$$

where $\Delta\dot{V}(s(t))$ and $\Delta\boldsymbol{\varphi}$, respectively, are the change in $\dot{V}(s(t))$ and $\boldsymbol{\varphi}$.

Applying the chain rule yields the following:

$$\begin{aligned} \frac{\partial \dot{V}(s(t))}{\partial \boldsymbol{\varphi}} &= \frac{\partial \dot{V}(s(t))}{\partial \hat{u}_{sTTRCA}} \frac{\partial \hat{u}_{MSTT2FCA}}{\partial \boldsymbol{\varphi}} \\ &= \frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}_{MSTT2FCA}} \frac{\partial \hat{u}_{MSIT2FCA}}{\partial \boldsymbol{\varphi}} \end{aligned} \tag{42}$$

From (24) and (41), it is obtained that:

$$\frac{\partial \dot{V}(s(t))}{\partial \boldsymbol{\varphi}} = \frac{1}{2}s(t)\frac{\partial \hat{u}_{MSIT2FCA}}{\partial \boldsymbol{\varphi}} = \frac{1}{2}s(t)\boldsymbol{\zeta}_{\boldsymbol{\varphi}}(t) \tag{43}$$

From (24)–(33), $\Delta\boldsymbol{\varphi}$ can be represented as

$$\Delta\boldsymbol{\varphi} = -\hat{\eta}_z\frac{\partial s(t)\dot{s}(t)}{\partial \boldsymbol{\varphi}} = -\frac{1}{2}\hat{\eta}_{\boldsymbol{\varphi}}s(t)\boldsymbol{\zeta}_{\boldsymbol{\varphi}}(t) \tag{44}$$

Using (41), (43), and (44), it is obtained that

$$\begin{aligned}\Delta\dot{V}(s(t)) &= \left[\frac{\partial \dot{V}(s(t))}{\partial \boldsymbol{\varphi}}\right]^T \Delta\boldsymbol{\varphi} \\ &= \left[\frac{1}{2}s(t)\boldsymbol{\zeta}_{\varphi}(t)\right]^T\left[-\frac{1}{2}\hat{\eta}_{\varphi}s(t)\boldsymbol{\zeta}_{\varphi}(t)\right] \\ &= -\frac{1}{2}\hat{\eta}_{\varphi}(s(t))^2\left(\boldsymbol{\zeta}_{\varphi}(t)\right)^2 \end{aligned} \tag{45}$$

In (45), if $\hat{\eta}_{\varphi}$ is chosen as a positive value, then $\Delta\dot{V}(s(t))$ is a negative semi-definite, so $\boldsymbol{s}(t)$ is bounded. The control law $\hat{\boldsymbol{u}}_{MSIT2FCA}$ ensures that the system is stable. For the proposed MSIT2FCA, the network parameters are tuned online using the derived adaptive laws that are shown in (24)-(33). Correct learning rates must be used for these adaptive laws because they have a significant effect on the control performance. This study uses a MGWO algorithm to determine the optimal learning rates. The controller's parameters can then be learned to achieve quick convergence for the control systems. The detail of the MGWO algorithm is presented in the following section.

### C. THE MGWO

According to [42], the grey wolves' hunting mechanisms are based on the leadership strategy, in which each grey wolf can adjust their position based on the best position, the second-best position, and the third-best position in the swarm. In this study, in order to improve the searchability of the GWO, an MGWO based on enhancing the random searching positions and memoing the best solution is proposed.

The formula for updating the position of the MGWO is:

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right|;\ \vec{D}_\beta = \left|\vec{C}_2.\vec{X}_\beta - \vec{X}\right|;\ \vec{D}_\delta = \left|\vec{C}_3.\vec{X}_\delta - \vec{X}\right| \tag{46}$$

$$\vec{X}_1 = \left|\vec{X}_\alpha - \vec{A}_1.\vec{D}_\alpha\right|;\ \vec{X}_2 = \left|\vec{X}_\beta - \vec{A}_2.\vec{D}_\beta\right|;\ \vec{X}_3 = \left|\vec{X}_\delta - \vec{A}_3.\vec{D}_\delta\right| \tag{47}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} + F\left[\phi_1;\ \phi_2;\ldots;\ \phi_{n_d}\right] \tag{48}$$

where $\vec{X}_\alpha$, $\vec{X}_\beta$, $\vec{X}_\delta$ respectively represent the best position, the second-best position and the third-best position for the grey wolves in the swarm; $\vec{X}$ is the current position of a grey wolf; $\vec{C}_1$, $\vec{C}_2$, $\vec{C}_3$ and $\vec{A}_1$, $\vec{A}_2$, $\vec{A}_3$ are the coefficient vectors. $\vec{D}_\alpha$, $\vec{D}_\beta$, $\vec{D}_\delta$ respectively denote the distance vectors between

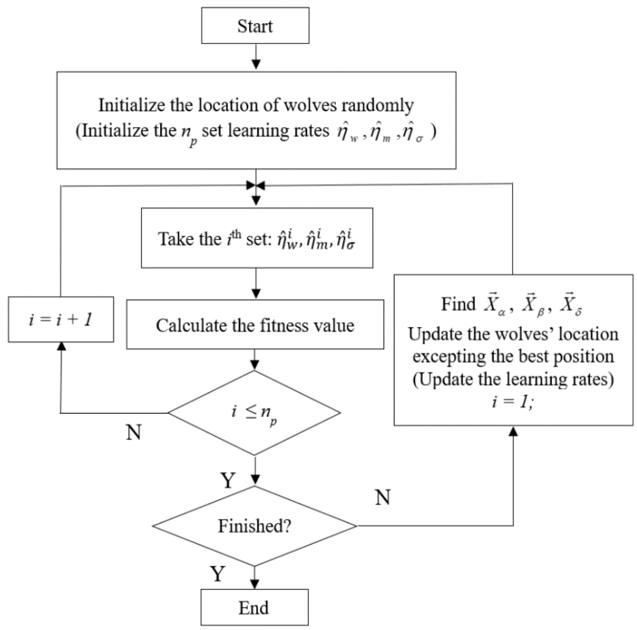


**FIGURE 3. The flowchart for the MGWO algorithm.**

the $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ value, and the current position of a grey wolf. $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ are the position vectors for calculating the next position of a grey wolf $\vec{X}(t+1)$, and $\phi_{n_d}$ is the random coefficient factor in [0, 1] for the $n_d$ dimension of the searching space. $F$ is the variable that is used to adjust the value of $\phi_{n_d}$, which is given by:

$$F = 0.05^*\left(satlin\left(\|e(t)\|^4\right)\right) \tag{49}$$

The coefficient vectors $\vec{A}$ and $\vec{C}$ are written as:

$$\vec{A} = 2ar_1 - a;\quad \vec{C} = 2r_2 \tag{50}$$

where $a$ is a number that linearly decreases from 2 to 0 over the course of the iterations and $r_1$ and $r_2$ are random numbers in [0, 1].

The fitness function is chosen as:

$$fitness = (e_1(t) + e_2(t) + e_3(t))^2 \tag{51}$$

As shown in Eq (48), the random coefficient factor $\phi_{n_d}$ will make the grey wolves in the swarm moving more random in $n_d$ demension of searching space. In (50), vectors $\vec{A}$ and $\vec{C}$ contains random values $r_1$ and $r_2$. According to [42], swarm solutions tend to diverge from the prey when $|A| > 1$ (exploration operation) and converge towards the prey when $|A| < 1$ (exploitation operation). The same can be said for vector C, if $C > 1$, the stochastically emphasize operation will occurred, contrarily, if $C < 1$, the deemphasize operation will occurred.

Unlike the original GWO in [42], this study's MGWO develops the terms $F$ and $\phi_{n_d}$ in the formula for updating the grey wolf's position. These terms increase the searching performance for the GWO and avoid convergence to a local optimum. For this study's MGWO, the best position in each iteration is stored and is only updated when a better position

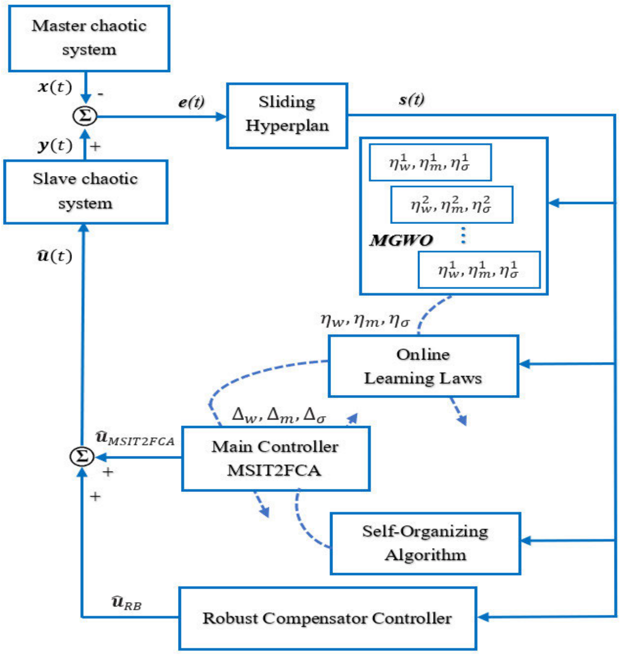


**FIGURE 4.** The scheme for the chaotic synchronization systems.

becomes apparent in the next iteration. The MGWO is used to optimize the learning rates $\hat{\eta}_w$, $\hat{\eta}_m$, $\hat{\eta}_\sigma$ for the adaptive laws. Fig. 3 shows the flowchart for the MGWO algorithm.

### D. STRUCTURE LEARNING FOR THE MSIT2FCA

In terms of the network structure for MSIT2FCA, the number of layers in the association memory space significantly affects the system's performance. If there are too few layers, the system may not be relevant to all cases, especially if there is a broad range of inputs. If there are too many layers, the computation time is excessive. This section details the self-organizing algorithm that autonomously determines the structure of the MSIT2FCA controller. The mechanism to generate a new MF or to prune an unused MF uses the membership grade that, corresponds to the current input signal.

The condition for generating a new AMF is:

$$\mu^i_{\max} < G_{th} \tag{52}$$

where $G_{th}$ is the generating threshold and $\mu^i_{\max}$ is the maximum membership grade of the $i^{\text{th}}$ input, which is written as:

$$\mu^i_{\max} = max \left[\mu_{i11}, \ldots, \mu_{i1n_k}, \mu_{i21}, \ldots, \mu_{i2n_k}, \ldots, \mu_{in_j1}, \ldots, \mu_{in_jn_k}\right] \tag{53}$$

where the average MF, $\mu_{ijk}$, is:

$$\mu_{ijk} = \frac{\underline{\mu}_{ijk} + \overline{\mu}_{ijk}}{2} \tag{54}$$

The parameters for the new AMF are defined as

$$\left[\overline{m}^l_{ijk}, \underline{m}^l_{ijk}, \underline{m}^r_{ijk}, \overline{m}^r_{ijk}\right] = \left[(I_i - 2\alpha), (I_i - \alpha), (I_i + \alpha), (I_i + 2\alpha)\right] \tag{55}$$

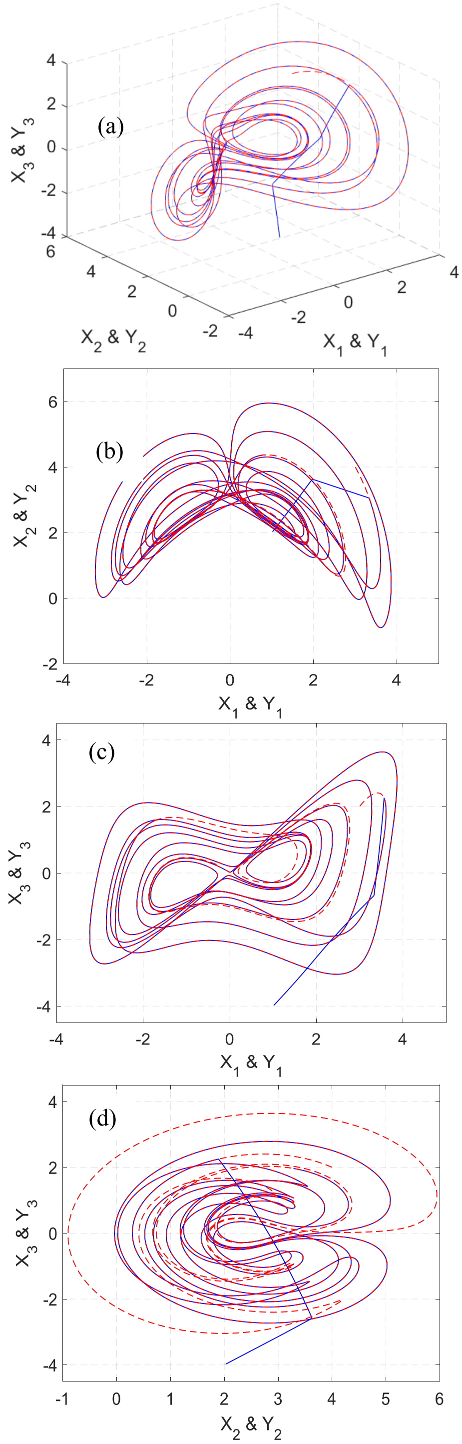


**FIGURE 5.** The synchronization of the chaotic satellite systems using the MSIT2FCA controller in example 1. (a) Trajectories projected on the three-dimensional plane. (b) Trajectories projected on the $x_1$–$x_2$ plane. (c) Trajectories projected on the $x_1$–$x_3$ plane. (d) Trajectories projected on the $x_2$–$x_3$ plane.

$$\left[\underline{\sigma}^l_{ijk}, \overline{\sigma}^l_{ijk}, \underline{\sigma}^r_{ijk}, \overline{\sigma}^r_{ijk}\right] = \left[(\sigma_{init} - 2\beta), (\sigma_{init} - \beta), (\sigma_{init} + \beta), (\sigma_{init} + 2\beta)\right] \tag{56}$$

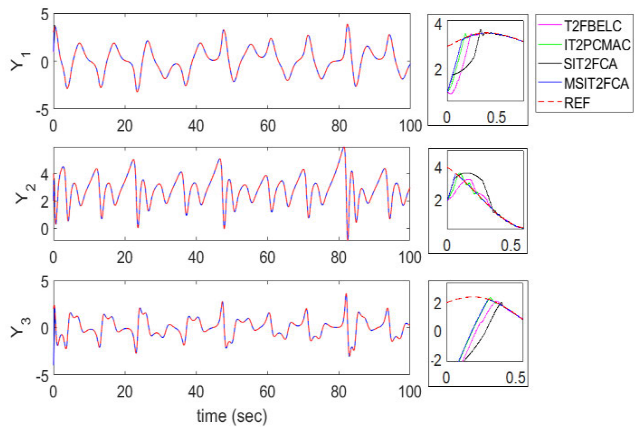


**FIGURE 6.** The trajectory and synchronization results using various controllers in example 1.

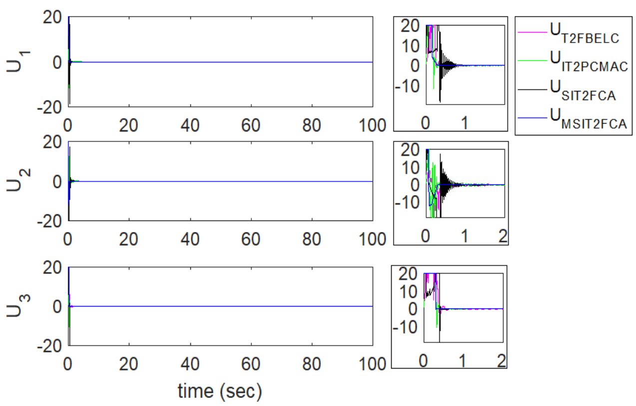


**FIGURE 7.** The control signals using various controllers in example 1.

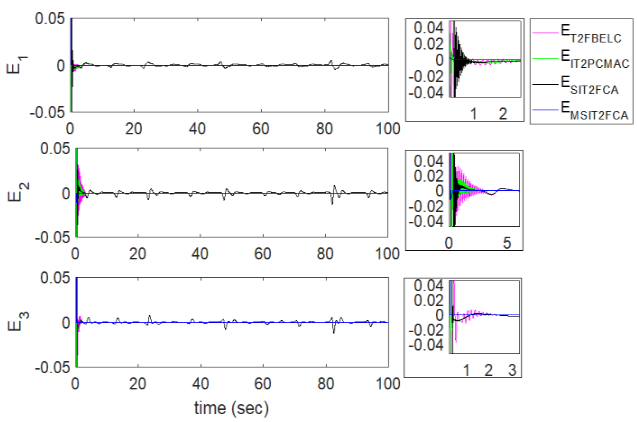


**FIGURE 8.** The tracking errors using various controllers in example 1.

where $\alpha$ and $\beta$ respectively represent the uncertainty in the mean and the uncertainty in the variance and $\sigma_{init}$ is the initial value of the variance.

The condition for deleting an unused AMF is:

$$\mu^i_{\min} < D_{th} \tag{57}$$

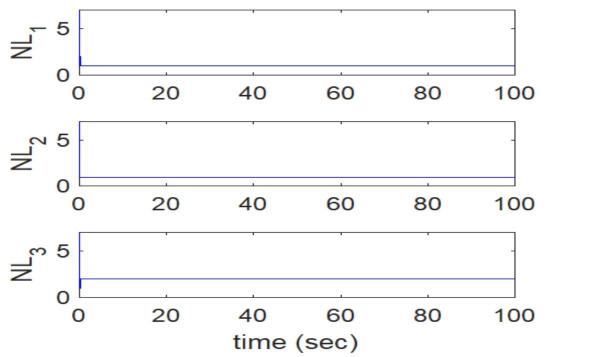


**FIGURE 9.** The layer adjustment of the MSIT2FCA controllers in example 1.

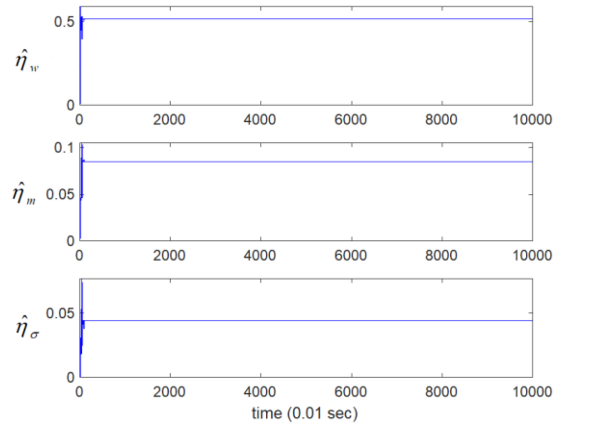


**FIGURE 10.** The online adjustment of the learning rates using the MGWO in example 1.

where $D_{th}$ is the deletion threshold and $\mu^i_{\min}$ is the minimum membership grade of the $i^{\text{th}}$ input, which is written as:

$$\mu^i_{\min} = min \left[\mu_{i11}, \ldots, \mu_{i1n_k}, \mu_{i21}, \ldots, \mu_{i2n_k}, \ldots, \mu_{in_j1}, \ldots, \mu_{in_jn_k}\right] \tag{58}$$

This autonomously increasing and decreasing MF is used to optimize the structure of the MSIT2FCA controller.

## IV. ILLUSTRATIVE EXAMPLES

The estimation controller, $\hat{\boldsymbol{u}}(t)$, in (17) is used to ensure that a slave system $\boldsymbol{y}(t)$ follows a master system $\boldsymbol{x}(t)$ in order to contain the tracking errors $\boldsymbol{e}(t)$ in a small bounded region. Fig. 4 shows the scheme for the chaotic synchronization systems that use the MSIT2FCA controller.

As in [3], the initial conditions for both the master and slave satellite systems are $\boldsymbol{x}(0) = [3,\ 4,\ 2]$, $\boldsymbol{y}(0) = \lfloor 1,\ 2,\ -4 \rfloor$, the principal moments of inertia are $I_x = 3$, $I_y = 2$, $I_z = 1$, the external disturbances are $\boldsymbol{d}(t) = [\cos \pi t,\ 0.5 \cos t,\ 1.5 \cos 2t]^T$, and the system uncertainties are $\Delta \boldsymbol{f}(t) = [0.8y_1,\ 0.8y_2,\ 0.8y_3]^T$. In order to decrease the computational burden, the maximum number of layers is limited to five and the minimum number of layers is one. The MGWO algorithm is used to optimize three learning rates $\hat{\eta}_w, \hat{\eta}_m, \hat{\eta}_\sigma$ for the adaptive laws, then the dimension of the searching space is chosen as $n_d = 3$; the population

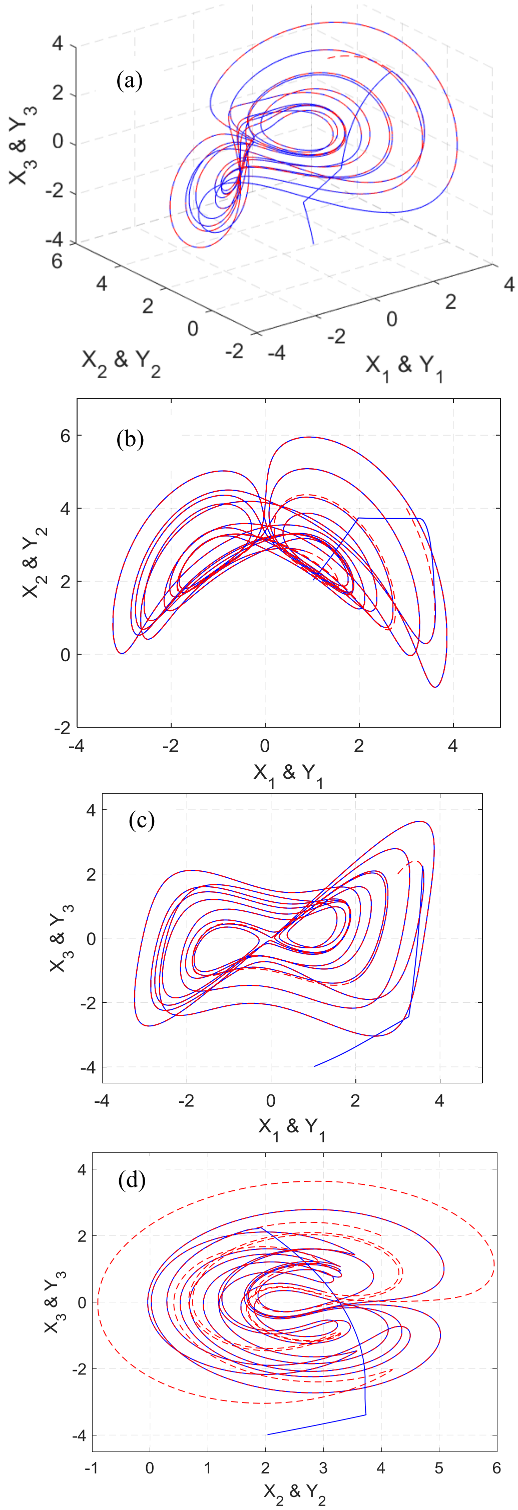


**FIGURE 11.** **The synchronization of the chaotic satellite systems using the MSIT2FCA controller in example 2. (a) Trajectories projected on the three-dimensional plane. (b) Trajectories projected on the x1–x2 plane. (c) Trajectories projected on the x1–x3 plane. (d) Trajectories projected on the x2–x3 plane.**

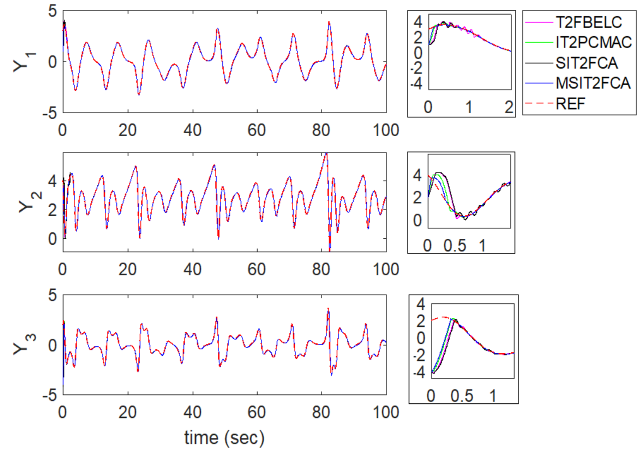


**FIGURE 12.** **The trajectory and synchronization results using various controllers in example 2.**

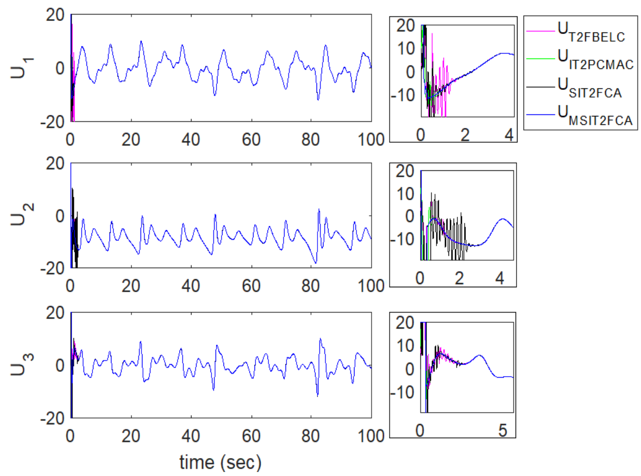


**FIGURE 13.** **The control signals using various controllers in example 2.**

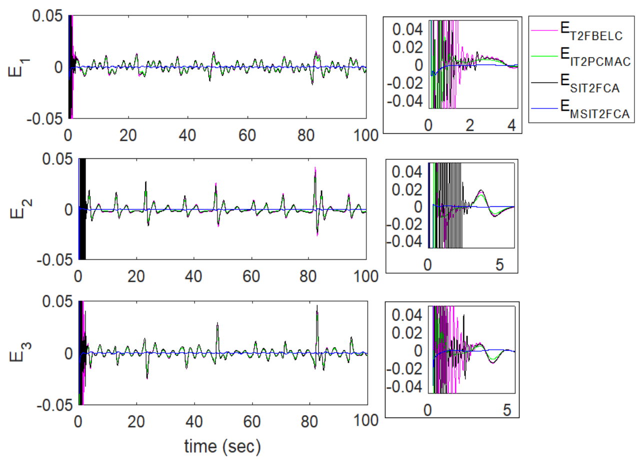


**FIGURE 14.** **The tracking errors using various controllers in example 2.**

size is chosen as $n_p = 20$. The parameter values for the self-organizing algorithm are $G_{th} = 0.15$, $D_{th} = 0.01$, $\alpha = 0.02$, $\sigma_{init} = 0.3$, $\beta = 0.05$, and the sampling time is 0.001 s.

The performance of the synchronization system is calculated using the root mean square error (RMSE), which is written as:

$$RMSE = \sqrt{\frac{1}{n_s}\sum_{s=1}^{n_s}\left((e_{1s})^2 + (e_{2s})^2 + (e_{3s})^2\right)} \tag{59}$$

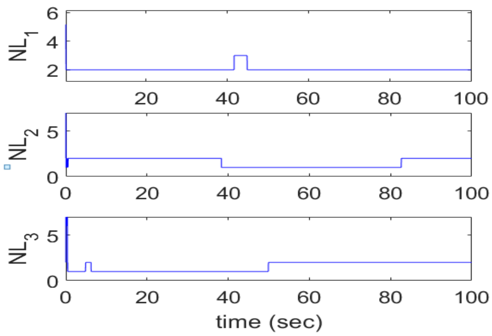


**FIGURE 15.** The layer adjustment of the MSIT2FCA controllers in example 2.

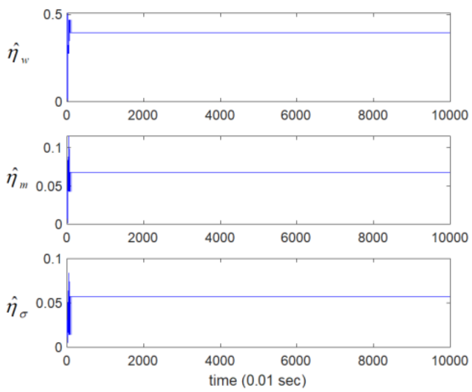


**FIGURE 16.** The online learning rate adjustment using the MGWO in example 2.

where $n_s$ is the number of samples and $e_{1s}, e_{2s}, e_{3s}$ is the tracking error for the $s^{th}$ sample.

Two examples of chaotic satellite synchronization are demonstrated in the simulation. The first simulation involves no external disturbances or system uncertainties (shown in Figs. 5–10). Fig. 5 shows the process of synchronization for the three-dimensional chaotic satellite systems using the MSIT2FCA controller. Fig. 6 shows the trajectories and synchronization results for various controllers. Figs. 7 and 8 respectively show the control signals and tracking errors for a synchronized chaotic satellite system. Fig. 9 shows how the number of layers in the MSIT2FCA controller is adjusted using a self-organizing algorithm. Fig. 10 shows the adjustment of the online learning rate using the MGWO. The synchronization tracking errors in the chaotic satellite system rapidly converge to zero and synchronization is achieved quickly.

The simulation results for the synchronization of a chaotic satellite systems with external disturbances and system uncertainties are illustrated in Figs. 11–16. Fig. 11 shows the synchronization for the three-dimensional chaotic satellite systems using the MSIT2FCA controller. Fig. 12 shows the trajectories and synchronization results for various controllers. Figs. 13 and 14 respectively show the control signals and tracking errors for a synchronized chaotic satellite system. Fig. 15 shows the adjustment of the number of layers for the MSIT2FCA controllers using the self-organizing algorithm. Fig. 16 shows the adjustment of the online learning rate using the MGWO. The synchronization tracking errors for the chaotic satellite system quickly converge to zero, so synchronization is achieved quickly. This example demonstrates that the proposed controller addresses external disturbances and system uncertainties.

**TABLE 1.** Comparison results in RMSE synchronization of the chaotic satellite system.

| | Computation time (s) | Example 1 | Example 2 |
|---|---|---|---|
| T2FBELC | 0.0183 | 0.3409 | 0.4431 |
| IT2PCMAC | 0.0172 | 0.2841 | 0.3503 |
| SIT2FCA | 0.0165 | 0.3552 | 0.4629 |
| MSIT2FCA | 0.0196 | 0.2675 | 0.3014 |

The simulation results in Figs. 9 and 15 verify that the MSIT2FCA controllers quickly construct suitable network layers using the self-organizing algorithm. Figs. 10 and 16 show that the MGWO allows the proposed controller to rapidly converge to a suitable value so it synchronizes a chaotic satellite system with the smallest tracking errors. A comparison of the RMSE for the proposed controller and other control methods is shown in Table 1. It is seen that the proposed MSIT2FCA controller ensure better synchronization than a type-2 fuzzy-brain emotional-learning controller (T2FBELC) [31], an interval type-2 Petri CMAC (IT2PCMAC) [52], or the proposed controller without the MGWO algorithm (SIT2AFC). The simulation program code is given in [55].

## V. CONCLUSION

This study determines the optimal network structure and the optimal learning rate for a MSIT2FCA using a self-organizing algorithm and a MGWO. The adaptive laws for the online updating of network parameters are derived using the gradient descent method. An AMF is used to increase the learning capability and the flexibility of the proposed network. Two examples of simulated synchronization of chaotic satellites verify the effectiveness of the proposed system. The comparison shows that the proposed controller addresses external disturbances and system uncertainties to give the best synchronization performance. Future study will estimate the generation and deletion thresholds that achieve the best control performance.

## REFERENCES

[1] S. Farzami Sarcheshmeh, R. Esmaelzadeh, and M. Afshari, "Chaotic satellite synchronization using neural and nonlinear controllers," *Chaos, Solitons Fractals*, vol. 97, pp. 19–27, Apr. 2017.

[2] S. Hamel, A. Bouzeriba, and A. Boulkroune, "Function vector synchronization based on fuzzy control for uncertain chaotic systems with dead-zone nonlinearities," *Complexity*, vol. 21, no. S1, pp. 234–249, 2016.

[3] A. Khan and S. Kumar, "T-S fuzzy modeling and predictive control and synchronization of chaotic satellite systems," *Int. J. Model. Simul.*, vol. 39, no. 3, pp. 203–213, Jul. 2019.

[4] A. Senouci and A. Boukabou, "Predictive control and synchronization of chaotic and hyperchaotic systems based on a T–S fuzzy model," *Math. Comput. Simul.*, vol. 105, pp. 62–78, Nov. 2014.

[5] S. Hamel and A. Boulkroune, "Adaptive fuzzy control-based projective synchronization scheme of uncertain chaotic systems with input nonlinearities," in *Proc. Int. Conf. Electr. Eng. Control Appl.* Cham, Switzerland: Springer, 2016, pp. 45–59.

[6] D. Sadaoui, A. Boukabou, N. Merabtine, and M. Benslama, "Predictive synchronization of chaotic satellites systems," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 9041–9045, Jul. 2011.

[7] A. Khan and S. Kumar, "Measure of chaos and adaptive synchronization of chaotic satellite systems," *Int. J. Dyn. Control*, vol. 7, no. 2, pp. 536–546, Jun. 2019.

[8] Y. Farid and T. V. Moghaddam, "Generalized projective synchronization of chaotic satellites problem using linear matrix inequality," *Int. J. Dyn. Control*, vol. 2, no. 4, pp. 577–586, Dec. 2014.

[9] C.-M. Lin, T.-T. Huynh, and T.-L. Le, "Adaptive TOPSIS fuzzy CMAC back-stepping control system design for nonlinear systems," *Soft Comput.*, vol. 23, no. 16, pp. 6947–6966, Aug. 2019, doi: 10.1007/s00500-018-3333-4.

[10] W. Fang, F. Chao, L. Yang, C.-M. Lin, C. Shang, C. Zhou, and Q. Shen, "A recurrent emotional CMAC neural network controller for vision-based mobile robots," *Neurocomputing*, vol. 334, pp. 227–238, Mar. 2019.

[11] C.-M. Lin, Y.-L. Liu, and H.-Y. Li, "SoPC-based function-link cerebellar model articulation control system design for magnetic ball levitation systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4265–4273, Aug. 2014.

[12] C.-C. Chung, T.-S. Chen, L.-H. Lin, Y.-C. Lin, and C.-M. Lin, "Bankruptcy prediction using cerebellar model neural networks," *Int. J. Fuzzy Syst.*, vol. 18, no. 2, pp. 160–167, Apr. 2016.

[13] C.-M. Lin and T.-L. Le, "WCMAC-based control system design for nonlinear systems using PSO," *J. Intell. Fuzzy Syst.*, vol. 33, no. 2, pp. 807–818, Jul. 2017.

[14] J.-G. Wang, S.-C. Tai, and C.-J. Lin, "The application of an interactively recurrent self-evolving fuzzy CMAC classifier on face detection in color images," *Neural Comput. Appl.*, vol. 29, no. 6, pp. 201–213, Mar. 2018.

[15] D. Zhou, M. Shi, F. Chao, C.-M. Lin, L. Yang, C. Shang, and C. Zhou, "Use of human gestures for controlling a mobile robot via adaptive CMAC network and fuzzy logic controller," *Neurocomputing*, vol. 282, pp. 218–231, Mar. 2018.

[16] C.-M. Lin and T.-Y. Chen, "Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1377–1384, Sep. 2009.

[17] C.-H. Lee, F.-Y. Chang, and C.-M. Lin, "An efficient interval type-2 fuzzy CMAC for chaos time-series prediction and synchronization," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 329–341, Mar. 2014.

[18] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, 1975.

[19] 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.

[20] Y. Pan, M. J. Er, T. Sun, B. Xu, and H. Yu, "Adaptive fuzzy PD control with stable $H^\infty$ tracking guarantee," *Neurocomputing*, vol. 237, pp. 71–78, May 2017.

[21] R. P. A. Gil, Z. C. Johanyak, and T. Kovacs, "Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation," *Int. J. Artif. Intell.*, vol. 16, no. 1, pp. 20–40, Mar. 2018.

[22] J. Soto, O. Castillo, P. Melin, and W. Pedrycz, "A new approach to multiple time series prediction using MIMO fuzzy aggregation models with modular neural networks," *Int. J. Fuzzy Syst.*, vol. 21, no. 5, pp. 1629–1648, Jul. 2019.

[23] F. Gaxiola, P. Melin, F. Valdez, J. R. Castro, and O. Castillo, "Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO," *Appl. Soft Comput.*, vol. 38, pp. 860–871, Jan. 2016.

[24] O. Castillo, J. R. Castro, P. Melin, and A. Rodriguez-Diaz, "Application of interval type-2 fuzzy neural networks in non-linear identification and time series prediction," *Soft Comput.*, vol. 18, no. 6, pp. 1213–1224, Jun. 2014.

[25] H. K. Lam, F. H. F. Leung, and P. K. S. Tam, "Stable and robust fuzzy control for uncertain nonlinear systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 30, no. 6, pp. 825–840, Nov. 2000.

[26] R.-E. Precup and S. Preitl, "Development of fuzzy controllers with non-homogeneous dynamics for integral-type plants," *Electr. Eng.*, vol. 85, no. 3, pp. 155–168, Jul. 2003.

[27] O. Castillo, R. Martínez-Marroquín, P. Melin, F. Valdez, and J. 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.

[28] M. M. Zirkohi and T.-C. Lin, "Interval type-2 fuzzy-neural network indirect adaptive sliding mode control for an active suspension system," *Nonlinear Dyn.*, vol. 79, no. 1, pp. 513–526, Jan. 2015.

[29] I. Eyoh, R. John, and G. De Maere, "Interval type-2 A-intuitionistic fuzzy logic for regression problems," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 2396–2408, Aug. 2018.

[30] M. Pratama, G. Zhang, M. J. Er, and S. Anavatti, "An incremental type-2 meta-cognitive extreme learning machine," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 339–353, Feb. 2017.

[31] T.-L. Le, T.-T. Huynh, and C.-M. Lin, "Adaptive filter design for active noise cancellation using recurrent type-2 fuzzy brain emotional learning neural network," *Neural Comput. Appl.*, pp. 1–10, Jul. 2019, doi: 10.1007/s00521-019-04366-8.

[32] C.-M. Lin, M.-S. Yang, F. Chao, X.-M. Hu, and J. Zhang, "Adaptive filter design using type-2 fuzzy cerebellar model articulation controller," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2084–2094, Oct. 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7312475

[33] K. Shiev, S. Ahmed, N. Shakev, and A. V. Topalov, "Trajectory control of manipulators using an adaptive parametric type-2 fuzzy CMAC friction and disturbance compensator," in *Novel Applications of Intelligent Systems*. Springer, 2016, pp. 63–82.

[34] C.-W. Chang, W.-R. Xiao, C.-C. Hsiao, S.-S. Chen, and C.-W. Tao, "A simplified interval type-2 fuzzy CMAC," in *Proc. Joint 17th World Congr. Int. Fuzzy Syst. Assoc., 9th Int. Conf. Soft Comput. Intell. Syst. (IFSA-SCIS)*. New York, NY, USA: IEEE Press, Jun. 2017, pp. 1–4.

[35] C.-M. Lin, V.-H. La, and T.-L. Le, "DC–DC converters design using a type-2 wavelet fuzzy cerebellar model articulation controller," *Neural Comput. Appl.*, pp. 1–13, Oct. 2018, doi: 10.1007/s00521-018-3755-z.

[36] T.-L. Le, "Fuzzy C-Means clustering interval type-2 cerebellar model articulation neural network for medical data classification," *IEEE Access*, vol. 7, pp. 20967–20973, 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8631033

[37] T. Zhao, P. Li, and J. Cao, "Self-organising interval type-2 fuzzy neural network with asymmetric membership functions and its application," *Soft Comput.*, vol. 23, no. 16, pp. 7215–7228, Aug. 2019.

[38] T. Ozen and J. M. Garibaldi, "Effect of type-2 fuzzy membership function shape on modelling variation in human decision making," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, vol. 2, Jun. 2004, pp. 971–976.

[39] H. Hellendoorn and C. Thomas, "Defuzzification in fuzzy controllers," *J. Intell. Fuzzy Syst.*, vol. 1, no. 2, pp. 109–123, 1993.

[40] C.-H. Lee, T.-W. Hu, C.-T. Lee, and Y.-C. Lee, "A recurrent interval type-2 fuzzy neural network with asymmetric membership functions for nonlinear system identification," in *Proc. IEEE Int. Conf. Fuzzy Syst., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 1496–1502.

[41] P. Kumar, R. Prasad, V. N. Mishra, D. K. Gupta, A. Choudhary, and P. K. Srivastava, "Artificial neural network with different learning parameters for crop classification using multispectral datasets," in *Proc. Int. Conf. Microw., Opt. Commun. Eng. (ICMOCE)*, Dec. 2015, pp. 204–207.

[42] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[43] L. Rodríguez, O. Castillo, J. Soria, P. Melin, F. Valdez, C. I. Gonzalez, G. E. Martinez, and J. Soto, "A fuzzy hierarchical operator in the grey wolf optimizer algorithm," *Appl. Soft Comput.*, vol. 57, pp. 315–328, Aug. 2017.

[44] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "A grey wolf optimizer for optimum parameters of multiple PI controllers of a grid-connected PMSG driven by variable speed wind turbine," *IEEE Access*, vol. 6, pp. 44120–44128, 2018.

[45] H. Faris, S. Mirjalili, and I. Aljarah, "Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2901–2920, Oct. 2019, doi: 10.1007/s13042-018-00913-2.

[46] A. Saxena, R. Kumar, and S. Das, "$\beta$-Chaotic map enabled grey wolf optimizer," *Appl. Soft Comput.*, vol. 75, pp. 84–105, Feb. 2019.

[47] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, "Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification," *Neural Comput. Appl.*, pp. 1–20, Jul. 2019, doi: 10.1007/s00521-019-04368-6.

[48] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble grey wolf optimizer and its application to feature selection," *Appl. Soft Comput.*, vol. 76, pp. 16–30, Mar. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1568494618306793

[49] Y. Pan and M. J. Er, "Enhanced adaptive fuzzy control with optimal approximation error convergence," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1123–1132, Dec. 2013.

[50] A. Goli, A. Aazami, and A. Jabbarzadeh, "Accelerated cuckoo optimization algorithm for capacitated vehicle routing problem in competitive conditions," *Int. J. Artif. Intell.*, vol. 16, no. 1, pp. 88–112, 2018.

[51] R.-E. Precup and R.-C. David, *Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems*. London, U.K.: Butterworth, 2019.

[52] T.-L. Le, "Self-organizing recurrent interval type-2 Petri fuzzy design for time-varying delay systems," *IEEE Access*, vol. 7, pp. 10505–10514, 2019.

[53] 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," *Int. J. Fuzzy Syst.*, vol. 21, no. 8, pp. 2524–2541, Nov. 2019.

[54] H. Liu, Y. Pan, S. Li, and Y. Chen, "Adaptive fuzzy backstepping control of fractional-order nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2209–2217, Aug. 2017.

[55] *Synchronize Chaotic Satellite Systems*. Accessed: Mar. 17, 2020. [Online]. Available: https://drive.google.com/drive/folders/1HAc_SZP2kx5u022WQtjDEwoKkkFBX3aP?usp=sharing

**TIEN-LOC LE** (Member, IEEE) was born in Vietnam, in 1985. He received the B.S. degree in electronics and telecommunication engineering from Lac Hong University, Vietnam, in 2009, the M.S. degree in electrical engineering from the HCMC University of Technology and Education, Vietnam, in 2012, and the Ph.D. degree in electrical engineering from Yuan Ze University, Taoyuan, Taiwan, in January 2018. From February 2018 to December 2018, he was a Postdoctoral Research Fellow of the Department of Electrical Engineering, Yuan Ze University. He is currently a Postdoctoral Research Fellow of the Faculty of Mechanical and Aerospace Engineering, Sejong University, Seoul, South Korea. He is also a Lecturer with the Faculty of Mechatronics and Electronics, Lac Hong University. His research interests include intelligent control systems, fuzzy neural networks, type-2 fuzzy neural networks, and cerebellar model articulation controller.

**TUAN-TU HUYNH** (Member, IEEE) was born in Ho Chi Minh City, Vietnam, in 1982. He received the B.S. degree in electrical and electronics from the Department of Electrical and Electronics Engineering, Ho Chi Minh University of Technology and Education, Vietnam, in 2005, the M.S. degree in automation from the Ho Chi Minh City University of Transport, Vietnam, in 2010, and the Ph.D. degree in electrical engineering from Yuan Ze University, Taoyuan, Taiwan, in 2018. He is currently a Research Fellow of the Department of Electrical Engineering, Yuan Ze University. He is also a Lecturer with the Department of Electrical, Electronics, and Mechanical Engineering, Lac Hong University, Vietnam. His research interests include MCDM, fuzzy logic control, neural networks, cerebellar model articulation controller, brain emotional learning-based intelligent controller, deep learning, and intelligent control systems.

**SUNG-KYUNG HONG** received the B.S. and M.S. degrees in mechanical engineering from Yonsei University, Seoul, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree from Texas A&M University, College Station, in 1998. From 1989 to 2000, he was with the Unmanned Aerial Vehicle System Division and the Flight Dynamics and Control Laboratory, Agency for Defense Development (ADD), South Korea. He is currently a Full Professor with the Department of Aerospace Engineering, Sejong University, South Korea. His research interests include fuzzy logic control, inertial sensor applications, and flight control systems for unmanned vehicles.