IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 29, 2018, accepted December 18, 2018, date of publication December 21, 2018,
date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889226

Self-Organizing Recurrent Interval Type-2 Petri Fuzzy Design for Time-Varying Delay Systems

TIEN-LOC LE, (Member, IEEE)
Department of Electrical Engineering, Yuan Ze University, Taoyuan 320, Taiwan
Department of Electrical Electronic and Mechanical Engineering, Lac Hong University, Biên Hòa 810000, Vietnam
e-mail: tienloc@saturn.yzu.edu.tw

This work was supported in part by the Vietnam National Foundation for Science and Technology Development, and in part by the Yuan Ze University, Taoyuan, Taiwan.

ABSTRACT This paper presents a design of a self-organizing recurrent interval type-2 Petri fuzzy controller (SORIT2PFC) for controlling the time-varying delay systems. The control system comprises of a main controller and a compensation controller; the SORIT2PFC as the main controller is used to imitate an ideal controller, and the simple fuzzy compensator controller is used to eliminate the residual error. By using a self-organizing algorithm, the structure of the proposed network can automatically achieve optimal construction. Recurrent neural network and fuzzy Petri nets are applied to improve system performances and to reduce computational burden. Online tune adaptive laws of the proposed controller are derived by implementing the gradient descent method and the Lyapunov stability theorem. Finally, the control efficacy and effectiveness of the proposed controller are verified by the numerical simulations of the time-varying delay systems.

INDEX TERMS Type-2 fuzzy system, Petri-fuzzy, recurrent neural network, self-organizing algorithm, time-varying delay systems.

I. INTRODUCTION

Type-1 fuzzy sets (T1FSs) were introduced by Zadeh [1] in 1965, and have been attracting many researchers for the past decades. Because T1FSs use precise sets in the membership function, they can not cover the uncertainties which come from the internal and external disturbances well [2]. Therefore, in 1975, Zadeh provided type-2 fuzzy sets (T2FSs), which have the uncertainty in the membership function [3]. Because the T2FSs are complex in computation, in 2000, Liang and Mendel introduced a simple way to implement the T2FSs, This is called the interval type-2 fuzzy logic system (IT2FLS) [4]. Recently, many researchers have demonstrated the advantages of IT2FLS [5]–[12]. In 2015, Zhou *et al.* [11] provided an interval type-2 fuzzy control for nonlinear discrete-time systems with time-varying delays. Later in 2018, Huang *et al.* [12] presented an interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum. However, a fuzzy neural network with fixed structure cannot achieve a better performance when the system has uncertainties and disturbances [13]. Therefore, recently, many studies have applied the self-organizing algorithm to obtain a suitable network size for the fuzzy neural network structure [14]–[17]. In 2014, Lin *et al.* [15] introduced a self-organizing interval type-2 fuzzy neural network for radar emitter identification. In 2017, Lin and Le [16] provided a PSO-self-organizing interval type-2 fuzzy neural network for antilock braking systems. Following that, in 2018, Sabahi [17] presented the self-organizing fuzzy neural network applied to impedance force control. However, most of these methods are complex, and their control performance can be further improved. Thus, our proposed control system will apply the recurrent neural network and the Petri Nets to achieve better learning performance.

The recurrent neural network (RNN) was first introduced by Hopfield [18] in 1984. RNN includes the time delays term in its structure and is composed of many massively connected simple neurons that can operate concurrently [19]. By applying the RNN, the system can have previous information and hence the network performance can be further improved [20]. In the past decades, many studies have applied the feature of the RNN to their designed network structure [21]–[25]. In 2016, Bao and Zheng [22] proposed a discrete-time recurrent neural network with discontinuous activation functions. Also in 2016, Lin *et al.* [23] introduced a recurrent fuzzy neural cerebellar model articulation network fault-tolerant control of six-phase permanent magnet synchronous motor position servo drive. In 2017, Mansoori *et al.* [25] provided an efficient recurrent neural network model for solving fuzzy non-linear programming problems.

Petri Nets (PNs) are a graphic modeling method, which was introduced by Peterson [26] in 1981. After that, PNs have been widely applied to models and used to analyze such discrete event systems as communication, manufacturing, and transportation systems [27]. Since Looney proposed the fuzzy Petri nets (FPNs), which is the combination of fuzzy and PNs, it has attracted many researchers' attention in the field of artificial intelligence [28]. The FPNs have some advantages such as reducing the computational burden and providing a faster learning ability [29], [30]. Recently, many studies have applied FPNs in various fields such as control problem, system identification, prediction and classification [30]–[34]. In 2017, Bibi *et al.* [31] introduced a Petri type 2 fuzzy neural networks approximator for adaptive control of uncertain nonlinear systems. Also in 2017, Rosdi *et al.* [32] provided an FPN-based classification method for speech intelligibility detection of children with speech impairments. Following that, in 2018, Zhu *et al.* [34] presented model-based fault identification of discrete event systems using partially observed Petri nets. Compare to the previous studies in [30]–[34], this research proposes an effective control method, which combines the advantages of the interval type-2 fuzzy network, the recurrent network, the Petri nets, and the self-organizing algorithm. It leads the proposed control system to better cover uncertainties, is effective in reducing the computational burden, and makes it easy to design the initial network parameters.

Motivated by the aforementioned discussion, in this study, a self-organizing recurrent interval type-2 Petri fuzzy controller (SORIT2PFC) is proposed for the time-varying delay systems. The term time-varying delay in this model can be considered as system uncertainties; therefore the SORIT2PFC is proposed to better cope with these uncertainties. In the control scheme, the SORIT2PFC is the main controller, and the simple fuzzy acts as the compensator controller which is used to eliminate the residual error. The main contributions of this study include: (1) The development of a SORIT2PFC network which has the adaptive laws for online updating parameters; (2) The self-organizing algorithm is applied to autonomously construct the size of the network controller; (3) The Petri layer with a dynamic threshold is applied to eliminate unsuitable rules and reduce the computational burden; (4) The recurrent network is applied to help the system have previous information and improve system performances; (5) The stability of system is proven using the Lyapunov function approach; (6) The numerical simulation results of the time-varying delay systems are conducted to illustrate the effectiveness of the proposed control system.

The remainder of this paper is organized as follows: The design of the SORIT2PFC control system is presented in Section 2. The details of the parameter learning and the compensator controller are shown in Section 3. The numerous control results of the time-varying delay systems are shown in Section 4. Finally, the conclusions are given in Section 5.

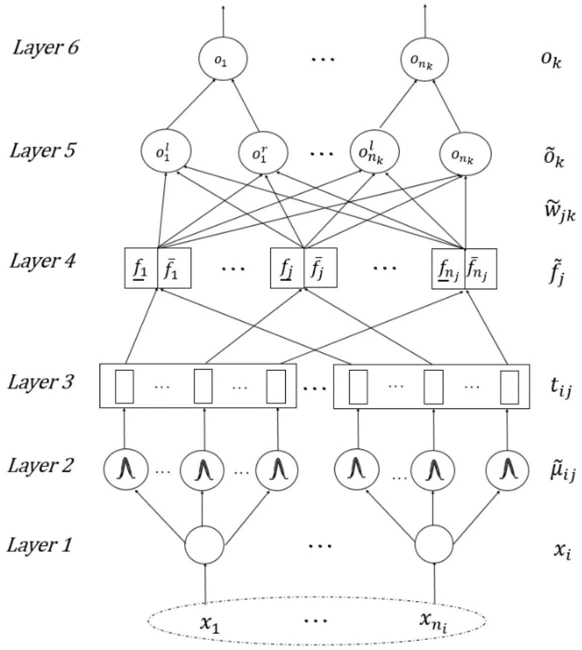


FIGURE 1. The architecture of the SORIT2PFC.

II. STRUCTURE OF THE SELF-ORGANIZING RIT2PFC

A. THE ARCHITECTURE OF RIT2PFC

The fuzzy rule of the proposed controller RIT2PFC is defined as follows:

$$\begin{aligned}
&Rule\ j: IF\ x_1\ is\ \tilde{\mu}_{1j}\ and\ x_2\ is\ \tilde{\mu}_{2j}, \ldots, and\ x_i\ is\ \tilde{\mu}_{ij}\\
&\quad Then\ \tilde{w}_{jk} = \left[\underline{w}_{jk}\ \bar{w}_{jk}\right]\\
&\quad for\ i = 1, 2, \ldots i, \ldots, n_i; j = 1, 2, \ldots, j, \ldots n_j;\\
&\quad k = 1, 2, \ldots, j, \ldots n_k
\end{aligned} \tag{1}$$

where n_i is the input dimension, n_j is the number of the firing node, and n_k is the output dimension. The fuzzy set for the i^{th} input and the j^{th} membership function are denoted by $\tilde{\mu}_{ij}$, the output weight for connecting the j^{th} firing node and k^{th} output is denoted by $\tilde{w}_{jk}$.

The structure of the recurrent interval type-2 Petri fuzzy neural network (RIT2PFNN) with the self-organizing structure is shown in Fig. 1. This structure can be described by the functions and tasks of six layers as follow:

1) THE INPUT LAYER

This layer consists of a vector input signal $\boldsymbol{x} = [x_1, x_2, \ldots x_i, \ldots x_{n_i}]^T \Re^{n_i}$ that is directly transferred to the input of the next layer.

2) THE MEMBERSHIP FUNCTION LAYER

Each node in this layer performs a type-2 Gaussian membership function (T2GMF) combined with the recurrent term.

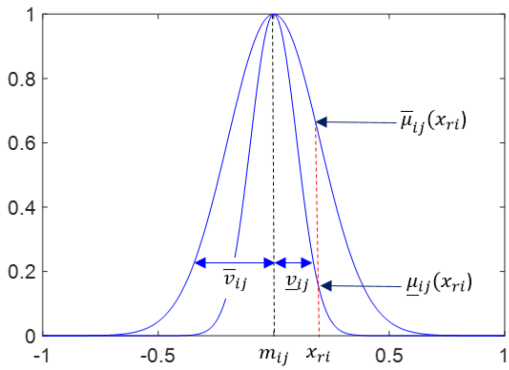


FIGURE 2. Interval type-2 Gaussian membership function.

The outputs of this layer are represented as

$$\underline{\mu}_{ij} = \exp\left\{-\frac{1}{2}\left(\frac{\underline{x}_{ri} - m_{ij}}{\underline{v}_{ij}}\right)^2\right\} \tag{2}$$

$$\bar{\mu}_{ij} = \exp\left\{-\frac{1}{2}\left(\frac{\bar{x}_{ri} - m_{ij}}{\bar{v}_{ij}}\right)^2\right\} \tag{3}$$

where $\tilde{\mu}_{ij} \in \left[\underline{\mu}_{ij}, \bar{\mu}_{ij}\right]$ is the membership grade, which can be obtained by the recurrent input and the T2GMF as shown in Fig. 2; m_{ij} and $\tilde{v}_{ij} \in \left[\underline{v}_{ij}, \bar{v}_{ij}\right]$ are the mean and the uncertain variance of the T2GMF, respectively; $\underline{x}_{ri}$ and $\bar{x}_{ri}$ are the recurrent inputs, which can be given by

$$\begin{aligned} \underline{x}_{ri}(t) &= x_i(t) + r_{ij}\underline{\mu}_{ij}(t-1) \\ \bar{x}_{ri}(t) &= x_i(t) + r_{ij}\bar{\mu}_{ij}(t-1) \end{aligned} \tag{4}$$

3) THE PETRI LAYER

In this layer the Petri nets act as a transition layer, which is used to produce the tokens, and the fuzzy laws are selected based on suitable fired nodes.

$$t_{ij} = \begin{cases} 1, & \mu_{ij} \geq g_{th} \\ 0, & \mu_{ij} < g_{th} \end{cases} \tag{5}$$

where t_{ij} and g_{th} are the transition nodes and the dynamic threshold value. The equation for choosing g_{th} is dependent on the corresponding error as

$$g_{th} = \frac{\varphi \exp(-\psi E)}{1 + \exp(-\psi E)} \tag{6}$$

where φ and ψ are positive constants for adjusting the Petri threshold, $E = \frac{1}{2}e^2$ is the energy function, and e is the tracking error, which will be introduced in the next sub-section.

4) THE FIRING LAYER

This layer performs the product operation to obtain the fuzzy firing strength. The membership function layer has an interval value $\left[\underline{\mu}_{ij}, \bar{\mu}_{ij}\right]$, therefore the firing layer also has an interval value $\tilde{f}_j = \left[\underline{f}_j, \bar{f}_j\right]$

$$\underline{f}_j = \prod_{i=1}^{n_i} \underline{\mu}_{ij} \quad \text{and} \quad \bar{f}_j = \prod_{i=1}^{n_i} \bar{\mu}_{ij} \tag{7}$$

5) THE PRE-OUTPUT LAYER

The outputs of this layer can be obtained by using the firing nodes and the connecting weight vector, which are defined as

$$o_k^l = \frac{\sum_{j=1}^M f_j^l \underline{w}_{jk}}{\sum_{j=1}^M f_j^l} \quad \text{and} \quad o_k^r = \frac{\sum_{j=1}^M f_j^r \bar{w}_{jk}}{\sum_{j=1}^M f_j^r} \tag{8}$$

where $\tilde{w}_{jk} = \left[\underline{w}_{jk} \bar{w}_{jk}\right]$ is the connecting weight, which is used to connect the j^{th} firing node and k^{th} pre-output node. And f_l^i, f_r^i are the firing strengths, which are given by

$$f_j^l = \begin{cases} \bar{f}_j, & j \leq L \\ \underline{f}_j, & j > L \end{cases} \quad \text{and} \quad f_j^r = \begin{cases} \underline{f}_j, & j \leq R \\ \bar{f}_j, & j > R \end{cases} \tag{9}$$

where L and R are the left and right switch points, respectively. These values can be obtained using the KM algorithm [35].

6) THE OUTPUT LAYER

This layer performs the algebraic sum of the pre-output space o_l and o_r, and is obtained as

$$u_{\text{SORIT2PFC}}^k = o_k = \frac{\left(o_k^l + o_k^r\right)}{2} \tag{10}$$

B. SELF-ORGANIZING ALGORITHM FOR RIT2PFC

In designing the network structure, determining the number of rules significantly affects the system performance. A large number of rules will lead to a huge computation budget for the control system, and a smaller number of rules may not cover all the cases, especially when the input changes with a wide range [36]. Therefore, in this study, the self-organizing algorithm is applied to autonomously construct the network size of the proposed RIT2PFC controller. The flowchart for increasing and decreasing rules is shown in Fig. 3.

The condition for generating a new rule is given by

$$If\ \left(G_i < T_g\right) Then\ \{Generate\ a\ new\ T2GMF\ and\ new\ rule\} \tag{11}$$

where T_g is the preset threshold for generating a new rule, G_i is the maximum membership grade of i^{th} input, which can be obtained by

$$G_i = max\left[\mu_{i1}, \mu_{i2}, \ldots, \mu_{ij}, \ldots, \mu_{in_j}\right] \tag{12}$$

where $\mu_{ij} = \frac{\underline{\mu}_{ij} + \bar{\mu}_{ij}}{2}$

The initial parameters for the new type-2 Gaussian membership function are given by

$$\begin{aligned} m_{ij}^{M(t)+1} &= x_i(t) \\ \left[\underline{v}_{ij}^{M(t)+1}, \bar{v}_{ij}^{M(t)+1}\right] &= \left[v_{init} - \Delta v, v_{init} + \Delta v\right] \end{aligned} \tag{13}$$

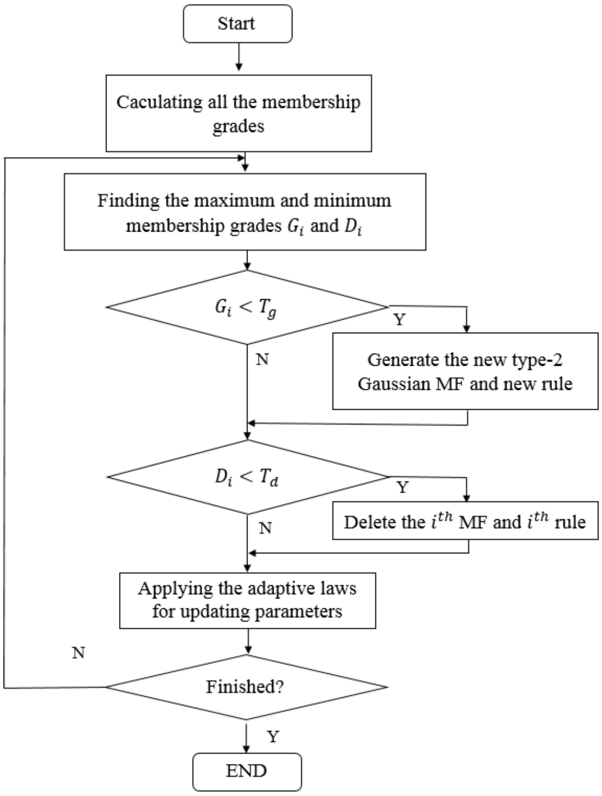

Start
Caculating all the membership grades
Finding the maximum and minimum membership grades G_i and D_i
$G_i < T_g$
Y
N
Generate the new type-2 Gaussian MF and new rule
$D_i < T_d$
Y
N
Delete the i^{th} MF and i^{th} rule
Applying the adaptive laws for updating parameters
N
Finished?
Y
END

FIGURE 3. Flowchart of the structure and parameter learning for SEIT2WCMAC.

where v_{init} and Δv are the initial values of the variance and half of the uncertain variance, respectively. The total number of existing rules at the k^{th} step is denoted by *M(k)*.

The condition for deleting an inappropriate rule is given by

$$If\ (D_i < T_d)\ Then\ \left\{Deleting\ i^{th}\ MF\ and\ its\ rules\right\} \tag{14}$$

where T_d is the preset threshold for deleting an inappropriate rule, D_i is the minimum membership grade of i^{th} input, which can be obtained by

$$D_i = min\left[\mu_{i1}, \mu_{i2}, \ldots, \mu_{ij}, \ldots, \mu_{in_j}\right] \tag{15}$$

By using this generating and deleting algorithm, the structure of the proposed RIT2PFC can automatically achieve optimal construction.

III. PARAMETER LEARNING AND COMPENSATOR CONTROLLER

A. PARAMETER LEARNING FOR SORIT2PFC

Consider a class of n^{th} order nonlinear systems described by:

$$\begin{aligned} x^{(n)}(t) &= f(\boldsymbol{x}(t)) + g(\boldsymbol{x}(t))\,u(t) + d(t) \\ &= f_0(\boldsymbol{x}(t)) + \Delta f(\boldsymbol{x}(t)) \\ &\quad + [g_0(\boldsymbol{x}(t)) + \Delta g(\boldsymbol{x}(t))]\,u(t) + d(t) \\ &= f_0(\boldsymbol{x}(t)) + g_0(\boldsymbol{x}(t))\,u(t) + \beta(\boldsymbol{x}(t)) \end{aligned} \tag{16}$$

In which, the lumped uncertainty term is denoted by $\beta(\boldsymbol{x}(t))$ and can be given as

$$\beta(\boldsymbol{x}(t)) = \Delta f(\boldsymbol{x}(t)) + \Delta g(\boldsymbol{x}(t))\,u(t) + d(t) \tag{17}$$

where $\boldsymbol{x} = \left[x, \dot{x}, \ddot{x}, \ldots x^{(n-1)}\right]^T$ is the state vector of system.

$x(t)$: the system output.

$u(t)$: the control input.

$d(t)$: the external disturbance.

$f(\boldsymbol{x})$, $g(\boldsymbol{x})$: the bounded nonlinear functions.

$f_0(\boldsymbol{x}(t))$, $g_0(\boldsymbol{x}(t))$: the nominal parts of the bounded nonlinear functions.

$\Delta f(\boldsymbol{x}(t))$, $\Delta g(\boldsymbol{x}(t))$: the unknown uncertainties of bounded nonlinear functions.

The tracking error is defined as

$$e(t) = x_d(t) - x(t) \tag{18}$$

where $x_d(t)$ is the reference signal.

From (16), the ideal controller can be defined as

$$u^*(t) = g_0^{-1}(\boldsymbol{x}(\mathrm{t}))\left[x_d^{(n)}(t) - f_0(\boldsymbol{x}(t)) - \beta(\boldsymbol{x}(t)) + \boldsymbol{K}^T\boldsymbol{e}(t)\right] \tag{19}$$

where $\boldsymbol{e}(t) = \left[e(t), \dot{e}(t), \ldots, e^{(n-1)}(t)\right]^T$ is the system tracking error vector; $\boldsymbol{K} = [k_n, \ldots, k_2, k_1]^T$ is the feedback gain vector.

Applying the ideal controller (19) into (16), the error dynamics is given as:

$$e^{(n)} + k_1 e^{(n-1)} + \ldots + k_n e = 0 \tag{20}$$

If the feedback gain vector $\boldsymbol{K}$ in (19) is selected according to the coefficients of a Hurwitz polynomial, then $\lim_{t\to\infty} e(t) = 0$. Since $\beta(\boldsymbol{x}(t))$ is unknown or perturbed, $u^*(t)$ is unobtainable.

Assume there exists an $\boldsymbol{u}^*_{SORIT2FPC}$ to approach the $u^*(t)$

$$\boldsymbol{u}^*(t) = \boldsymbol{u}^*_{SORIT2FPC}(\underline{w}^*, \bar{w}^*, m^*, \underline{v}^*, \bar{v}^*, t) + \varepsilon(t) \tag{21}$$

where $\underline{w}^*, \bar{w}^*, m^*, \underline{v}^*, \bar{v}^*$ are the optimal parameters for $\underline{w}, \bar{w}, m, \underline{v}, \bar{v}$; $\varepsilon(t)$ is the approximation error.

Since the optimal parameters in (21) cannot be obtained exactly, we design the estimation controller as

$$\hat{u}(t) = \hat{\boldsymbol{u}}_{SORIT2FPC}(\underline{w}^*, \bar{w}^*, m^*, \underline{v}^*, \bar{v}^*, t) + \hat{u}_F(t) \tag{22}$$

where $\hat{\underline{w}}, \hat{\bar{w}}, \hat{m}, \hat{\underline{v}}, \hat{\bar{v}}$ are the estimation of $\underline{w}^*, \bar{w}^*, m^*, \underline{v}^*, \bar{v}^*$; $\hat{u}_F$ is the estimation of the fuzzy compensator controller.

To achieve better control performance, the high-order sliding surface from [37] and [38] is applied as

$$\begin{aligned} s(t) &= \sum_{l=0}^{l-1} \frac{(n-1)!}{l!\,(n-l-1)!}\left(\frac{\partial}{\partial t}\right)^{n-l-1} \lambda^l e \\ &= e^{(n-1)} + (n-1)\,\lambda e^{(n-2)} \\ &\quad + (n-2)\,\lambda^2 e^{(n-3)} \ldots + \lambda^{n-1} e \end{aligned} \tag{23}$$

where the slope of the sliding surface can be adjusted using the positive constant λ.

From (23), we obtained

$$\begin{aligned}\dot{s}(t) &= e^{(n)} + (n-1)\lambda e^{(n-1)} \\ &\quad + (n-2)\lambda^2 e^{(n-2)} \ldots + \lambda^{n-1} e^{(1)} \\ &= e^{(n)} + \boldsymbol{K}^T \boldsymbol{e} \end{aligned} \tag{24}$$

where $\boldsymbol{K} = \left[(n-1)\lambda, (n-2)\lambda^2, \ldots . \lambda^{n-1}\right]^T \in \Re^{n-1}$ is the positive gain vector. If the feedback gain vector $\boldsymbol{K}$ is selected according to the coefficients of a Hurwitz polynomial, then $\lim\limits_{t\to\infty} e(t) = 0$.

Defining the Lyapunov cost function as

$$V_1(s(t)) = \frac{1}{2}s^2(t) \tag{25}$$

Taking the derivative of (25) and using (16), (19), (22), (24), we obtained

$$\begin{aligned}\dot{V}_1(t) &= s(t)\dot{s}(t) = s(t)\left[e^{(n)} + \boldsymbol{K}^T\boldsymbol{e}\right] \\ &= s(t)[x_d^{(n)}(t) - (f_0(\boldsymbol{x}(t)) + g_0(\boldsymbol{x}(t)) \\ &\quad \times \left(\hat{u}_{SORIT2FPC}\left(\underline{\hat{w}}, \hat{\bar{w}}, \hat{m}, \underline{\hat{\sigma}}, \hat{\bar{\sigma}}, t\right) + \hat{u}_F(t)\right) \\ &\quad + \beta(\boldsymbol{x}(t))) + \boldsymbol{K}^T\boldsymbol{e}]\end{aligned} \tag{26}$$

Applying the chain rule and the gradient descent method, the adaptive laws for updating $\underline{\hat{w}}, \hat{\bar{w}}, \hat{m}, \underline{\hat{v}}, \hat{\bar{v}}$ can be expressed as

$$\begin{aligned}&\underline{\hat{w}}_{jk}(t+1) \\ &= \underline{\hat{w}}_{jk}(t) - \hat{\eta}_w \frac{\partial s(t)\dot{s}(t)}{\partial \underline{\hat{w}}_{jk}} \\ &= \underline{\hat{w}}_{jk}(t) - \hat{\eta}_w \left[\frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}_{SORIT2FPC}} \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial o^l_k} \frac{\partial o^l_k}{\partial \underline{\hat{w}}_{jk}}\right] \\ &= \underline{\hat{w}}_j(t) + \frac{1}{2}\hat{\eta}_w s(t) g_0(\boldsymbol{x}(t)) \frac{f^l_j}{\sum_{j=1}^M f^l_j}\end{aligned} \tag{27}$$

$$\begin{aligned}&\hat{\bar{w}}_{jk}(t+1) \\ &= \hat{\bar{w}}_{jk}(t) - \hat{\eta}_w \frac{\partial s(t)\dot{s}(t)}{\partial \hat{\bar{w}}_{jk}} \\ &= \hat{\bar{w}}_{jk}(t) - \hat{\eta}_w \left[\frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}_{SORIT2FPC}} \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial o^r_k} \frac{\partial o^r_k}{\partial \hat{\bar{w}}_j}\right] \\ &= \hat{\bar{w}}_j(t) + \frac{1}{2}\hat{\eta}_w s(t) g_0(\boldsymbol{x}(t)) \frac{f^r_j}{\sum_{j=1}^M f^r_j}\end{aligned} \tag{28}$$

$$\begin{aligned}&\hat{m}_{ij}(t+1) \\ &= \hat{m}_{ij}(t) - \hat{\eta}_m \frac{\partial s(t)\dot{s}(t)}{\partial \hat{m}_{ij}} \\ &= \hat{m}_{ij}(t) - \hat{\eta}_m \left[\frac{1}{2}\frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}^k_{SORIT2FPC}}\left(\frac{\partial o^l_k}{\partial f^l_j}\frac{\partial f^l_j}{\partial \hat{m}_{ij}} + \frac{\partial o^r_k}{\partial f^r_j}\frac{\partial f^r_j}{\partial \hat{m}_{ij}}\right)\right] \\ &= \hat{m}_{ij}(t) + \frac{1}{2}\hat{\eta}_m s(t) g_0(\boldsymbol{x}(t)) \\ &\quad \times \left(\frac{\left(\underline{w}_{jk} - o^l_k\right)}{\sum_{j=1}^M f^l_j}\frac{\partial f^l_j}{\partial \hat{m}_{ij}} + \frac{\left(\bar{w}_{jk} - o^r_k\right)}{\sum_{j=1}^M f^r_j}\frac{\partial f^r_j}{\partial \hat{m}_{ij}}\right)\end{aligned} \tag{29}$$

$$\begin{aligned}&\underline{\hat{v}}_{ij}(t+1) \\ &= \underline{\hat{v}}_{ij}(t) - \hat{\eta}_v \frac{\partial s(t)\dot{s}(t)}{\partial \underline{\hat{v}}_{ij}} \\ &= \underline{\hat{v}}_{ij}(t) - \hat{\eta}_v \left[\frac{1}{2}\frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}^k_{SORIT2FPC}}\left(\frac{\partial o^l_k}{\partial f^l_j}\frac{\partial f^l_j}{\partial \underline{\hat{v}}_{ij}} + \frac{\partial o^r_k}{\partial f^r_j}\frac{\partial f^r_j}{\partial \underline{\hat{v}}_{ij}}\right)\right] \\ &= \underline{\hat{v}}_{ij}(t) + \frac{1}{2}\hat{\eta}_v s(t) g_0(\boldsymbol{x}(t)) \\ &\quad \times \left(\frac{\left(\underline{w}_{jk} - o^l_k\right)}{\sum_{j=1}^M f^l_j}\frac{\partial f^l_j}{\partial \underline{\hat{v}}_{ij}} + \frac{\left(\bar{w}_{jk} - o^r_k\right)}{\sum_{j=1}^M f^r_j}\frac{\partial f^r_j}{\partial \underline{\hat{v}}_{ij}}\right)\end{aligned} \tag{30}$$

$$\begin{aligned}&\hat{\bar{v}}_{ij}(t+1) \\ &= \hat{\bar{v}}_{ij}(t) - \hat{\eta}_v \frac{\partial s(t)\dot{s}(t)}{\partial \hat{\bar{v}}_{ij}} \\ &= \hat{\bar{v}}_{ij}(t) - \hat{\eta}_v \left[\frac{1}{2}\frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}^k_{SORIT2FPC}}\left(\frac{\partial o^l_k}{\partial f^l_j}\frac{\partial f^l_j}{\partial \hat{\bar{v}}_{ij}} + \frac{\partial o^r_k}{\partial f^r_j}\frac{\partial f^r_j}{\partial \hat{\bar{v}}_{ij}}\right)\right] \\ &= \hat{\bar{v}}_{ij}(t) + \frac{1}{2}\hat{\eta}_v s(t) g_0(\boldsymbol{x}(t)) \\ &\quad \times \left(\frac{\left(\underline{w}_{jk} - o^l_k\right)}{\sum_{j=1}^M f^l_j}\frac{\partial f^l_j}{\partial \hat{\bar{v}}_{ij}} + \frac{\left(\bar{w}_{jk} - o^r_k\right)}{\sum_{j=1}^M f^r_j}\frac{\partial f^r_j}{\partial \hat{\bar{v}}_{ij}}\right)\end{aligned} \tag{31}$$

where $\hat{\eta}_w, \hat{\eta}_m, \hat{\eta}_v$ are the learning-rates for adjusting the adaptive convergence speed. From (7), the derivative term of f^l_j and f^r_j in (29) - (31) can be $\underline{f}^i$ or $\bar{f}^i$

$$\begin{aligned}\frac{\partial \underline{f}_j}{\partial \hat{m}_{ij}} &= \frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial \hat{m}_{ij}} = \underline{f}_j \frac{\underline{x}_{ri} - \hat{m}_{ij}}{\left(\underline{\hat{\sigma}}_{ij}\right)^2}; \\ \frac{\partial \bar{f}_j}{\partial \hat{m}_{ij}} &= \frac{\partial \bar{f}_j}{\partial \bar{\mu}_{ij}}\frac{\partial \bar{\mu}_{ij}}{\partial \hat{m}_{ij}} = \bar{f}_j \frac{\bar{x}_{ri} - \hat{m}_{ij}}{\left(\hat{\bar{v}}_{ij}\right)^2}\end{aligned} \tag{32}$$

$$\begin{aligned}\frac{\partial \underline{f}_j}{\partial \hat{\bar{v}}_{ij}} &= \frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial \hat{\bar{v}}_{ij}} = \underline{f}_j \frac{\left(\underline{x}_{ri} - \hat{m}_{ij}\right)^2}{\left(\hat{\bar{v}}_{ij}\right)^3}; \\ \frac{\partial \bar{f}_j}{\partial \hat{\bar{v}}_{ij}} &= \frac{\partial \bar{f}_j}{\partial \bar{\mu}_{ij}}\frac{\partial \bar{\mu}_{ij}}{\partial \hat{\bar{v}}_{ij}} = \bar{f}_j \frac{\left(\bar{x}_{ri} - \hat{m}_{ij}\right)^2}{\left(\hat{\bar{v}}_{ij}\right)^3}\end{aligned} \tag{33}$$

$$\begin{aligned}\frac{\partial \underline{f}_j}{\partial \underline{\hat{v}}_{ij}} &= \frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial \underline{\hat{v}}_{ij}} = \underline{f}_j \frac{\left(\underline{x}_{ri} - \hat{m}_{ij}\right)^2}{\left(\underline{\hat{v}}_{ij}\right)^3}; \\ \frac{\partial \bar{f}_j}{\partial \underline{\hat{v}}_{ij}} &= \frac{\partial \bar{f}_j}{\partial \bar{\mu}_{ij}}\frac{\partial \bar{\mu}_{ij}}{\partial \underline{\hat{v}}_{ij}} = \bar{f}_j \frac{\left(\bar{x}_{ri} - \hat{m}_{ij}\right)^2}{\left(\underline{\hat{v}}_{ij}\right)^3}\end{aligned} \tag{34}$$

Applying the adaptive laws derived in (27)-(31), the optimal parameters for the proposed controller SERIT2FPC can be obtained, and the control system can achieve the desired performance.

The convergence Analysis:

Taking the derivative of (25), yields:

$$\dot{V}(s(t)) = s(t)\dot{s}(t) \tag{35}$$

Defined

$$\boldsymbol{P}_{\boldsymbol{\chi}}(k) = \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \boldsymbol{\chi}} \text{ for } \boldsymbol{\chi} = \underline{\hat{\boldsymbol{w}}}, \bar{\hat{\boldsymbol{w}}}, \hat{\boldsymbol{m}}, \underline{\hat{\boldsymbol{v}}}, \bar{\hat{\boldsymbol{v}}} \tag{36}$$

where

$$\begin{aligned}
&\boldsymbol{P}_{\underline{\hat{\boldsymbol{w}}}}(t) \\
&= \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{\boldsymbol{w}}}} \\
&= \left[\frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{w}}_{11}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{w}}_{1n_k}}, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{w}}_{21}}, \right. \\
&\quad \left. \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{w}}_{2n_k}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{w}}_{n_j1}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{w}}_{n_in_j}} \right]
\end{aligned}$$

$$\begin{aligned}
&\boldsymbol{P}_{\bar{\hat{\boldsymbol{w}}}}(t) \\
&= \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{\boldsymbol{w}}}} \\
&= \left[\frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{w}}_{11}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{w}}_{1n_k}}, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{w}}_{21}}, \right. \\
&\quad \left. \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{w}}_{2n_k}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{w}}_{n_j1}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{w}}_{n_in_j}} \right]
\end{aligned}$$

$$\begin{aligned}
&\boldsymbol{P}_{\hat{\boldsymbol{m}}}(t) \\
&= \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \hat{\boldsymbol{m}}} \\
&= \left[\frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \hat{m}_{11}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \hat{m}_{1n_k}}, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \hat{m}_{21}}, \right. \\
&\quad \left. \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \hat{m}_{2n_k}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \hat{m}_{n_j1}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \hat{m}_{n_in_j}} \right]
\end{aligned}$$

$$\begin{aligned}
&\boldsymbol{P}_{\underline{\hat{\boldsymbol{v}}}}(t) \\
&= \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{\boldsymbol{v}}}} \\
&= \left[\frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{v}}_{11}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{v}}_{1n_k}}, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{v}}_{21}}, \right. \\
&\quad \left. \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{v}}_{2n_k}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{v}}_{n_j1}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \underline{\hat{v}}_{n_in_j}} \right]
\end{aligned}$$

$$\begin{aligned}
&\boldsymbol{P}_{\bar{\hat{\boldsymbol{v}}}}(t) \\
&= \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{\boldsymbol{v}}}} \\
&= \left[\frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{v}}_{11}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{v}}_{1n_k}}, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{v}}_{21}}, \right. \\
&\quad \left. \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{v}}_{2n_k}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{v}}_{n_j1}}, \ldots, \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \bar{\hat{v}}_{n_in_j}} \right]
\end{aligned}$$

Using the gradient descent method for (35), yields

$$\begin{aligned}
\dot{V}(s(t+1)) &= \dot{V}(s(t)) + \Delta\dot{V}(s(t)) \\
&\cong \dot{V}(s(t)) + \left[\frac{\partial \dot{V}(s(t))}{\partial \boldsymbol{\chi}}\right]^T \Delta\boldsymbol{\chi}
\end{aligned} \tag{37}$$

where $\Delta\dot{V}(s(k))$ and $\Delta\boldsymbol{\chi}$ are the changes in $\dot{V}(s(k))$ and $\boldsymbol{\chi}$, respectively.

Applying the chain rule, we obtained

$$\begin{aligned}
\frac{\partial \dot{V}(s(t))}{\partial \boldsymbol{\chi}} &= \frac{\partial \dot{V}(s(t))}{\partial \hat{u}^k_{SORIT2FPC}} \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \boldsymbol{\chi}} \\
&= \frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}^k_{SORIT2FPC}} \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \boldsymbol{\chi}}
\end{aligned} \tag{38}$$

Using (26), yields

$$\begin{aligned}
\frac{\partial \dot{V}(s(t))}{\partial \boldsymbol{\chi}} &= -s(t)\, g_0(\boldsymbol{x}(t)) \frac{\partial \hat{u}^k_{SORIT2FPC}}{\partial \boldsymbol{\chi}} \\
&= -s(t)\, g_0(\boldsymbol{x}(t))\, \boldsymbol{P}_{\boldsymbol{\chi}}(t)
\end{aligned} \tag{39}$$

From (27)-(31), we obtained

$$\Delta\boldsymbol{\chi} = -\hat{\eta}_{\boldsymbol{\chi}} \frac{\partial s(t)\dot{s}(t)}{\partial \boldsymbol{\chi}} = \hat{\eta}_{\boldsymbol{\chi}} s(t)\, g_0(\boldsymbol{x}(t))\, \boldsymbol{P}_{\boldsymbol{\chi}}(t) \tag{40}$$

Substituting (39), (40) into (37)

$$\begin{aligned}
&\Delta\dot{V}(s(t)) \\
&= \left[\frac{\partial \dot{V}(s(t))}{\partial \boldsymbol{\chi}}\right]^T \Delta\boldsymbol{\chi} \\
&= \left[-s(t)\, g_0(\boldsymbol{x}(t))\, \boldsymbol{P}_{\boldsymbol{\chi}}(t)\right]^T * \hat{\eta}_{\boldsymbol{\chi}} s(t)\, g_0(\boldsymbol{x}(t))\, \boldsymbol{P}_{\boldsymbol{\chi}}(t) \\
&= -s^2(t)\,(g_0(\boldsymbol{x}(t)))^2\, \hat{\eta}_{\boldsymbol{\chi}} \boldsymbol{P}_{\boldsymbol{\chi}}(t)
\end{aligned} \tag{41}$$

From (41), if $\hat{\eta}_{\boldsymbol{\chi}}$ is chosen as a positive value, then $\Delta\dot{V}(s(t)) < 0$. Consequently, the stability of the proposed SORIT2FPC control system can be guaranteed by the Lyapunov stability theorem.

B. THE FUZZY COMPENSATOR CONTROLLER

In this part, a fuzzy compensator is designed to deal with the approximation error in (21). In order to quickly scope the $\varepsilon(t)$, only three simple rules are used as follows:

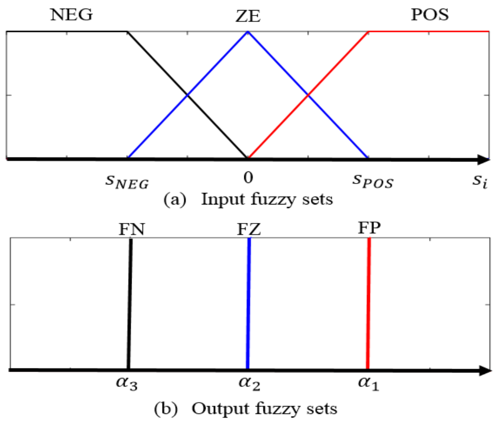
$$\begin{aligned}
&\textit{Rule } 1: \textit{If } s_i \textit{ is POS}, \quad \textit{then } u^i_F \textit{ is FP} \\
&\textit{Rule } 2: \textit{If } s_i \textit{ is ZE}, \quad \textit{then } u^i_F \textit{ is FZ} \\
&\textit{Rule } 1: \textit{If } s_i \textit{ is NEG}, \quad \textit{then } u^i_F \textit{ is FN}
\end{aligned} \tag{42}$$

where *POS, FP* are the positive membership functions for the input and output; *ZE, FZ* are the zero membership functions for the input and output; *NEG, FN* are the negative membership functions for input and output; s_i, u^i_F are the input and the fuzzy control signal at the i^{th} step.

The fuzzy rule membership functions for this compensator are shown in Fig. 4 in which the input membership functions *POS*, *ZE*, *NEG* are the triangular-typed functions, and the output membership functions *FP*, *FZ*, *FN* are the singletons.

The output of the fuzzy compensator controller can be obtained by the center-of-gravity method as

$$u_F = \frac{\sum\limits_{a=1}^{3} \alpha_a \beta_a}{\sum\limits_{a=1}^{3} \beta_a} = \alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_3\beta_3 \tag{43}$$



(a) Input fuzzy sets

(b) Output fuzzy sets

FIGURE 4. The fuzzy input and output membership functions for the compensator controller.

where $\beta_1, \beta_2, \beta_3$ and $\alpha_1, \alpha_2, \alpha_3$ are the firing strengths and the weights of fuzzy rules; $\beta_1, \beta_2, \beta_3$ are greater than or equal to zero, and based on the special case of the triangular membership function, we can obtain $\beta_1 + \beta_2 + \beta_3 = 1$. For simple computation, let $\alpha_1 = \hat{\alpha}$, $\alpha_2 = 0$ and $\alpha_3 = -\hat{\alpha}$. Therefore, when the input s_i going to the input membership function, only four cases can occur [39]:

Case 1: $s_i > s_{POS}$
Then

$$\beta_1 = 1, \quad \beta_2 = \beta_3 = 0 => u_F = \alpha_1 = \hat{\alpha} \tag{44}$$

Case 2: $0 < s_i < s_{POS}$
Then

$$\beta_1 > 0, \beta_2 \geq 1, \beta_3 = 0 => u_F = \alpha_1 \beta_1 = \hat{\alpha} \beta_1 \tag{45}$$

Case 3: $s_{NEG} < s_i \leq 0$
Then

$$\beta_1 = 0, \beta_2 > 0, \beta_3 \leq 1 => u_F = \alpha_3 \beta_3 = -\hat{\alpha} \beta_3 \tag{46}$$

Case 4: $s_i < s_{NEG}$
Then

$$\beta_1 = \beta_2 = 0, \beta_3 = 1 => u_F = \alpha_3 = -\hat{\alpha} \tag{47}$$

From (44)-(47), the general equation for computation fuzzy output is obtained:

$$u_F = \hat{\alpha} \left(\beta_1 - \beta_3 \right) \tag{48}$$

In order to achieve better control performance, the adaptive laws for updating $\hat{\alpha}$ from [39] is chosen as:

$$\dot{\hat{\alpha}} = s_i(t) \left(\beta_1 - \beta_3 \right) \tag{49}$$

IV. ILLUSTRATIVE EXAMPLES

Considering the following Henon system with time-varying delays in [40], which is expressed as:

$$\begin{aligned} x_1(k+1) &= -\left[c x_1(k) + (1-c) x_1(k-d(k)) \right]^2 \\ &\quad + 0.3 x_2(k) + 1.4 + u(k) x_2(k+1) \\ &= -c x_1(k) + (1-c) x_1(k-d(k)) \end{aligned} \tag{50}$$

where $c \in [0, 1]$ is the retarded coefficient, $d(k)$ is the time-varying state delay, which is randomly selected in the range $[3, 6]$. The signal $x_1(k - d(k))$ represents a signal $x_1(k)$ that is delayed $(k - d(k))$ steps. The $u(k)$ is the control signal, which is applied to control the system to achieve the stabilization state.

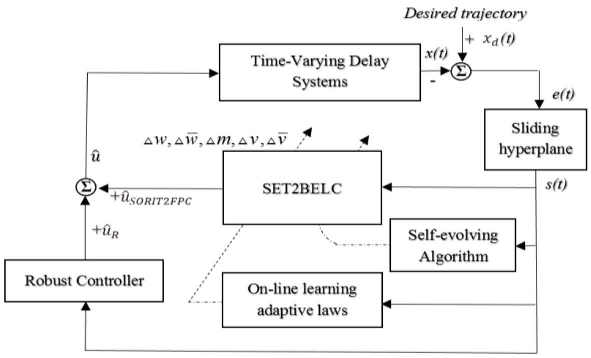


FIGURE 5. The block diagram of SORIT2FPC control system.

The scheme for the time-varying delays control system is shown in Fig. 5. The input for the proposed SORIT2FPC control system is the output of the high-order sliding surface and its derivative. The maximum number of T2GMF in each input is limited to 7 MFs. The states of the open-loop for Henon system are shown in Fig. 6. In this simulation, the values $c = 0.3$ and $c = 0.8$ are considered to illustrate the effectiveness of the proposed control system. The initial states of system are $x_1(k) = 1$, $x_2(k) = 0$. The initial parameters for the SORIT2FPC are chosen as $\eta_w = 0.1$, $\eta_m = 0.05$, $\eta_v = 0.05$, $m_{1j} = m_{2j} = [-0.300.3]$, $\underline{v}_{1j} = \underline{v}_{2j} = [0.10.10.1]$, $\bar{v}_{1j} = \bar{v}_{2j} = [0.20.20.2]$, $v_{init} = 0.5$, $\Delta v = 0.1$, $r_{ij} = 0.3$ $T_g = 0.1$ and $T_d = 0.03$.

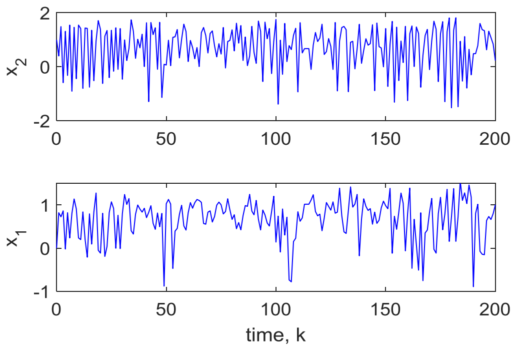


FIGURE 6. State of the open-loop for Henon system.

Figures 7 and 11 are the state of the closed-loop for the Henon system with $c = 0.3$ and $c = 0.8$, respectively and in which the blue-line and red-line are the outputs of the system using the proposed network with and without Petri nets, respectively. It can be seen that, when the Petri nets are applied, the control system can quickly achieve a stable state with a smaller tracking error than the controller without

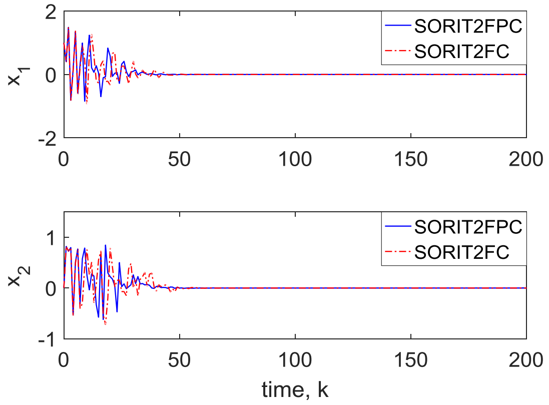


FIGURE 7. State of the closed-loop for Henon system with c = 0.3.

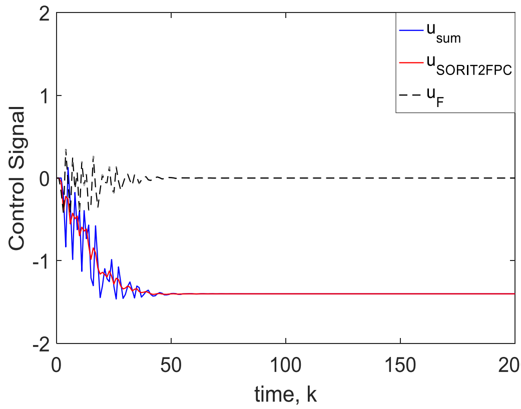


FIGURE 8. The control signals for Henon system with c = 0.3.

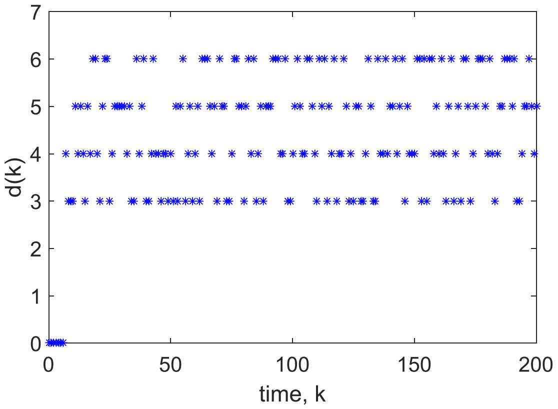


FIGURE 9. The random values d(k) for Henon system with c = 0.3.

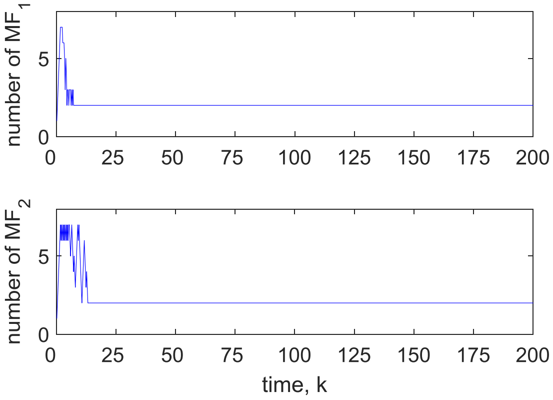


FIGURE 10. The number of MFs for SORIT2FPC controller with c = 0.3.

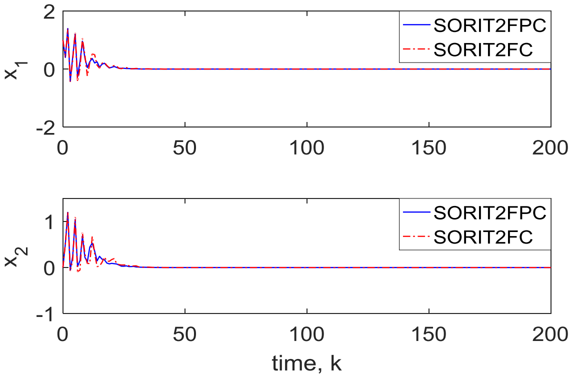


FIGURE 11. State of the closed-loop for Henon system with c = 0.8.

TABLE 1. Comparison results in RMSE.

	Computation time (s)	Case 1 c=0.3	Case 2 c=0.8
WCMAC [41]	0.0112	0.2169	0.1725
IT2FPC	0.0128	0.2145	0.1694
RIT2FPC	0.0132	0.2126	0.1673
SORIT2FC	0.0183	0.2093	0.1641
SORIT2FPC	0.0167	0.2079	0.1626

Petri nets. The control signals for the Henon system are shown in Fig. 8 and Fig. 12 in which, at the initial phase, the fuzzy compensator controller provides a large contribution for the total control effort to deal with the large tracking error, and then the SORIT2FPC controller plays the main controller to provide the desired control effort. The time-varying state delay term for two cases are shown in Fig. 9 and Fig. 13. The change in the number of MFs during the simulation time is given in Fig. 10 and Fig. 14 in which, at the initial phase, the self-organizing algorithm quickly generates the new MFs and new rules to scope the system errors, and quickly converges to the suitable rules after the system achieves the stabilization state. The root mean square errors (RMSE) comparison for both cases using different controllers are shown in Table 1. In both cases, it can be seen that the SORIT2FPC controller for Henon system can quickly achieve the stabilization state with the smallest tracking error. In Table 1, the comparison the results of the proposed network with and without Petri nets show that the SORIT2FPC can achieve better control performance. The Petri layer with a dynamic threshold can eliminate unsuitable rules, therefore, the computational time can be reduced and the system performance can be improved.

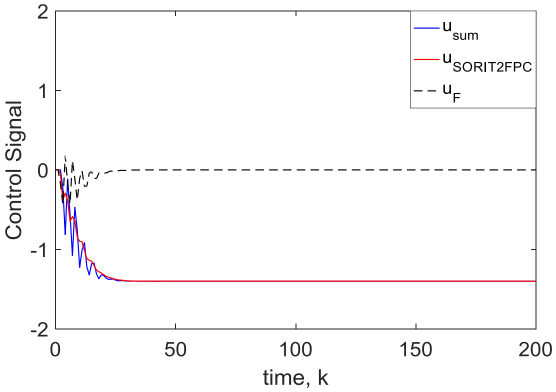


FIGURE 12. The control signals for Henon system with c = 0.8.

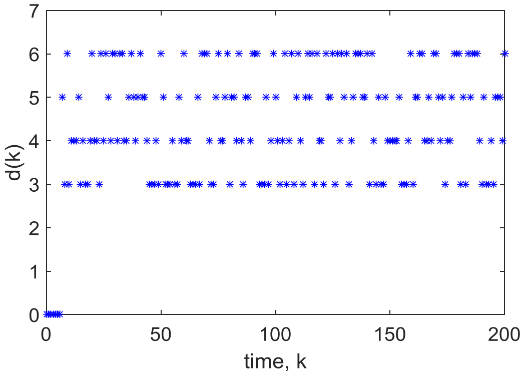


FIGURE 13. The random values d(k) for Henon system with c = 0.8.

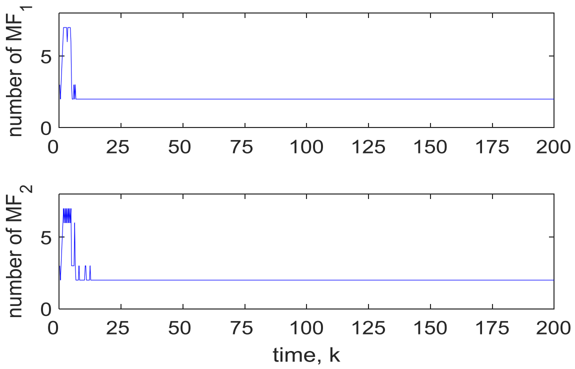


FIGURE 14. The number of MFs for SORIT2FPC controller with c = 0.8.

V. CONCLUSION

In this study, a novel adaptive SORIT2FPC controller combined with a fuzzy compensator controller is provided for controlling the time-varying delay systems. The major contribution of this work includes the development of a SORIT2PFC network which has the adaptive laws for online updating parameters; a self-organizing algorithm for autonomously achieves a suitable construction of SORIT2PFC network; the convergence of the system is proven by Lyapunov function analysis approach; the recurrent term and fuzzy Petri nets are applied to enhance system performance and to reduce the computational burden. Finally, the numerical simulation results of the time-varying delay systems have shown the effectiveness of the proposed control system. Choosing the threshold for generating and deleting the rules greatly affects the performance of the control systems. Therefore, applying the estimation method to estimate these thresholds will be our future work.

REFERENCES

[1] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[2] J. M. Mendel, "Type-2 fuzzy sets and systems: An overview [corrected reprint]," *IEEE Comput. Intell. Mag.*, vol. 2, no. 2, pp. 20–29, May 2007.

[3] 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.

[4] 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.

[5] 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.

[6] 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, 2015.

[7] 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.

[8] 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.

[9] X. Liu, F. Li, Y. Xie, C. Yang, and W. Gui, "Guaranteed cost control for descriptor type-2 fuzzy systems with stochastic delay distribution," *IEEE Access*, vol. 5, pp. 23637–23646, 2017.

[10] R. Sakthivel, R. Kavikumar, Y.-K. Ma, Y. Ren, and S. M. Anthoni, "Observer-based H_∞ repetitive control for fractional-order interval type-2 TS fuzzy systems," *IEEE Access*, vol. 6, pp. 49828–49837, 2018.

[11] Q. Zhou, D. Liu, Y. Gao, H.-K. Lam, and R. Sakthivel, "Interval type-2 fuzzy control for nonlinear discrete-time systems with time-varying delays," *Neurocomputing*, vol. 157, pp. 22–32, Jun. 2015.

[12] J. Huang, M. Ri, D. Wu, and S. Ri, "Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 2030–2038, Aug. 2018.

[13] C.-M. Lin, T.-L. Le, and T.-T. Huynh, "Self-evolving function-link interval type-2 fuzzy neural network for nonlinear system identification and control," *Neurocomputing*, vol. 275, pp. 2239–2250, Jan. 2018.

[14] 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.

[15] C.-M. Lin, Y.-M. Chen, and C.-S. Hsueh, "A Self-organizing interval type-2 fuzzy neural network for radar emitter identification," *Int. J. Fuzzy Syst.*, vol. 16, no. 1, pp. 20–30, 2014.

[16] C.-M. Lin and T.-L. Le, "PSO-self-organizing interval type-2 fuzzy neural network for antilock braking systems," *Int. J. Fuzzy Syst.*, vol. 19, no. 5, pp. 1362–1374, 2017.

[17] F. Sabahi, "Introducing validity into self-organizing fuzzy neural network applied to impedance force control," *Fuzzy Sets Syst.*, vol. 337, pp. 113–127, Apr. 2018.

[18] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. USA*, vol. 81, no. 10, pp. 3088–3092, 1984.

[19] X. Li, F. Li, X. Zhang, C. Yang, and W. Gui, "Exponential stability analysis for delayed semi-Markovian recurrent neural networks: A homogeneous polynomial approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6374–6384, Dec. 2018, doi: 10.1109/TNNLS.2018.2830789.

[20] Y.-T. Liu, Y.-Y. Lin, S.-L. Wu, C.-H. Chuang, and C.-T. Lin, "Brain dynamics in predicting driving fatigue using a recurrent self-evolving fuzzy neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 347–360, Feb. 2016.

[21] S.-I. Han and J.-M. Lee, "Recurrent fuzzy neural network backstepping control for the prescribed output tracking performance of nonlinear dynamic systems," *ISA Trans.*, vol. 53, no. 1, pp. 33–43, 2014.

[22] G. Bao and Z. Zeng, ''Global asymptotical stability analysis for a kind of discrete-time recurrent neural network with discontinuous activation functions,'' *Neurocomputing*, vol. 193, pp. 242–249, Jun. 2016.

[23] F.-J. Lin, I.-F. Sun, K.-J. Yang, and J.-K. Chang, ''Recurrent fuzzy neural cerebellar model articulation network fault-tolerant control of six-phase permanent magnet synchronous motor position servo drive,'' *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 153–167, Feb. 2016.

[24] F. F. El-Sousy, ''Intelligent mixed H_2/H_∞ adaptive tracking control system design using self-organizing recurrent fuzzy-wavelet-neural-network for uncertain two-axis motion control system,'' *Appl. Soft Comput.*, vol. 41, pp. 22–50, Apr. 2016.

[25] A. Mansoori, S. Effati, and M. Eshaghnezhad, ''An efficient recurrent neural network model for solving fuzzy non-linear programming problems,'' *Appl. Intell.*, vol. 46, no. 2, pp. 308–327, 2017.

[26] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 1981.

[27] M. Gao, M. Zhou, X. Huang, and Z. Wu, ''Fuzzy reasoning Petri nets,'' *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 33, no. 3, pp. 314–324, May 2003.

[28] C. G. Looney, ''Fuzzy Petri nets for rule-based decisionmaking,'' *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-18, no. 1, pp. 178–183, Jan./Feb. 1988.

[29] V. R. L. Shen, ''Reinforcement learning for high-level fuzzy Petri nets,'' *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 2, pp. 351–362, Apr. 2003.

[30] C.-M. Lin and H.-Y. Li, ''Dynamic Petri fuzzy cerebellar model articulation controller design for a magnetic levitation system and a two-axis linear piezoelectric ceramic motor drive system,'' *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 693–699, Mar. 2015.

[31] Y. Bibi, O. Bouhali, and T. Bouktir, ''Petri type 2 fuzzy neural networks approximator for adaptive control of uncertain non-linear systems,'' *IET Control Theory Appl.*, vol. 11, no. 17, pp. 3130–3136, Nov. 2017.

[32] F. Rosdi, S. S. Salim, and M. B. Mustafa, ''An FPN-based classification method for speech intelligibility detection of children with speech impairments,'' in *Proc. Soft Comput.*, 2017, pp. 1–18.

[33] P. Hansen, P. Franco, and S.-Y. Kim, ''Soccer ball recognition and distance prediction using fuzzy Petri nets,'' in *Proc. IRI*, Jul. 2018, pp. 315–322.

[34] G. Zhu, Z. Li, and N. Wu, ''Model-based fault identification of discrete event systems using partially observed Petri nets,'' *Automatica*, vol. 96, pp. 201–212, Oct. 2018.

[35] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[36] T.-L. Le, C.-M. Lin, and T.-T. Huynh, ''Self-evolving type-2 fuzzy brain emotional learning control design for chaotic systems using PSO,'' *Appl. Soft Comput.*, vol. 73, pp. 418–433, Dec. 2018.

[37] M. Manceur, N. Essounbouli, and A. Hamzaoui, ''Second-order sliding fuzzy interval type-2 control for an uncertain system with real application,'' *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 2, pp. 262–275, Apr. 2012.

[38] S. Ahmed, N. Shakev, A. Topalov, K. Shiev, and O. Kaynak, ''Sliding mode incremental learning algorithm for interval type-2 Takagi–Sugeno–Kang fuzzy neural networks,'' *Evolving Syst.*, vol. 3, no. 3, pp. 179–188, 2012.

[39] C.-M. Lin, M.-H. Lin, and C.-W. Chen, ''SoPC-based adaptive PID control system design for magnetic levitation system,'' *IEEE Syst. J.*, vol. 5, no. 2, pp. 278–287, Jun. 2011.

[40] L. Wu, X. Su, P. Shi, and J. Qiu, ''A new approach to stability analysis and stabilization of discrete-time T-S fuzzy time-varying delay systems,'' *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 41, no. 1, pp. 273–286, Feb. 2011.

[41] 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, 2017.

TIEN-LOC LE 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 Ho Chi Minh City University of Technology and Education, Vietnam, in 2012, and the Ph.D. degree in electrical engineering from Yuan Ze University, Taoyuan, Taiwan, in 2018, where he is currently a Postdoctoral Research Fellow with the Department of Electrical Engineering. He is also a Lecturer with Lac Hong University. His research interests include intelligent control systems, fuzzy neural networks, and cerebellar model articulation controller.