A TOPSIS based Self-Organizing Double Loop Recurrent Broad Learning System for Uncertain Nonlinear Systems

Huang, Wei-Zhong; Zhao, Yang; Hong, Wei-Bin; He, Hong-Rui; Chao, Fei; Yang, Longzhi; Lin, Chih-Min; Chang, Xiang; Shang, Changjing; Shen, Qiang

Published in:
2022 International Joint Conference on Neural Networks

DOI:
10.1109/IJCNN55064.2022.9892855

Publication date:
2022

Citation for published version (APA):
Huang, W.-Z., Zhao, Y., Hong, W.-B., He, H.-R., Chao, F., Yang, L., Lin, C.-M., Chang, X., Shang, C., & Shen, Q. (2022). A TOPSIS based Self-Organizing Double Loop Recurrent Broad Learning System for Uncertain Nonlinear Systems. In *2022 International Joint Conference on Neural Networks: IJCNN* (Proceedings of the International Joint Conference on Neural Networks; Vol. 2022-July). IEEE Press. https://doi.org/10.1109/IJCNN55064.2022.9892855

A TOPSIS based Self-Organizing Double Loop Recurrent Broad Learning System for Uncertain Nonlinear Systems

Huang, Wei-Zhong; Zhao, Yang; Hong, Wei-Bin; He, Hong-Rui; Chao, Fei; Yang, Longzhi; Lin, Chih-Min; Chang, Xiang; Shang, Changjing; Shen, Qiang

Published in:
2022 International Joint Conference on Neural Networks

DOI:
10.1109/IJCNN55064.2022.9892855

Publication date:
2022

Citation for published version (APA):
Huang, W-Z., Zhao, Y., Hong, W-B., He, H-R., Chao, F., Yang, L., Lin, C-M., Chang, X., Shang, C., & Shen, Q. (2022). A TOPSIS based Self-Organizing Double Loop Recurrent Broad Learning System for Uncertain Nonlinear Systems. In *2022 International Joint Conference on Neural Networks: IJCNN* IEEE Press. https://doi.org/10.1109/IJCNN55064.2022.9892855

A TOPSIS based Self-Organizing Double Loop Recurrent Broad Learning System for Uncertain Nonlinear Systems*

Wei-Zhong Huang*, Yang Zhao*, Wei-Bin Hong*, Hong-Rui He*, Fei Chao*§, Longzhi Yang†,
Chih-Min Lin‡, Xiang Chang§, Changjiang Shang§, and Qiang Shen§
*Department of Artificial Intelligence, Xiamen University, Xiamen, China 361005
Email: fchao@xmu.edu.cn
†Department of Computer and Information Sciences, Northumbria University, Newcastle, UK NE1 8ST
Email: longzhi.yang@northumbria.ac.uk
‡Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan 32003
Email: cml@saturn.yzu.edu.tw
§Department of Computer Science, Aberystwyth University, Aberystwyth, UK SY23 3DB
Email: {xic9, cns, qqs}@aber.ac.uk

***Abstract*—This study proposes an efficient intelligent control structure for uncertain nonlinear systems. The controller is implemented by a sliding mode control framework including a modified broad leaning network (BLS) with a double-loop recurrent structure. In addition, the proposed BLS involves a self-organizing mechanism to increase or decrease the size of the BLS. The technique for order of preference by similarity to ideal solution (TOPSIS) method is used to build the self-organizing mechanism. Moreover, two dynamic thresholds of TOPSIS are automatically determined according to the stability of the controller. One dynamic threshold is used to consider whether to retain or remove existing network neurons in the BLS; and the other is used to generate new neurons, so as to meet the requirements of different control states and save computing resources. To improve the network's dynamic characteristics, a double-loop recurrent structure is further introduced into the self-organizing BLS. The Lyapunov stability function is used to ensure the stability of the control system. The proposed controller is applied to the simulation control of a nonlinear chaotic system and a three-link robot manipulator. The experimental results show that the proposed controller can achieve better control performance against other network-based controllers. The source code of this work is placed at https://github.com/wzhuang-xmu/SODLRBLS**

***Index Terms*—Self-organizing network, broad learning system, double loop recurrent neural network, three-links robot manipulator**

I. Introduction

Nowadays, many studies have used learning methods to control uncertain nonlinear systems [1]–[3]; however, designing an efficient intelligent controller with excellent control performance is still a challenge. In particular, how counteracting non-linearity and uncertainty is the key to solving the challenge. Many studies suggest using the robust control algorithm as the main approach to controlling uncertain systems [4]–[7]. However, the sliding mode control (SMC) method can have better performance for nonlinear systems [8]–[10]. SMC can convert high-order systems into low-order systems and SMC is insensitive to parameter changes, capable of fast dynamic response, and able to suppress external disturbances [11].

Therefore, the SMC-based intelligent control systems are widely used in the control of nonlinear systems to obtain better learning ability and faster convergence speed. In addition, many SMC controllers involve various artificial neural networks to further improve their non-linear characteristics [12], [13]. For example, a fuzzy brain emotional learning network was embedded into an SMC control system for controlling different types of non-linear systems including robots [14]–[16]. In particular, the cerebellar model articulation controller network (CMAC) with a recurrent feedback loop has attracted good attention and led to promising results due to its structure's greater freedoms-of-design [17], [18].

However, current SMC controllers often suffer two challenges: (1) controllers consume too many computational resources, and (2) the accuracy of controllers is not high enough and their dynamic response characteristics are insufficient. For the first challenge, a self-organizing structure must be involved to efficiently and flexibly invoke computational resources; we thus find that the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) firstly proposed in [19] has been applied in many dynamic networks [20]–[22], whose results revealed that TOPSIS can use information contained in input data to rank and select candidate neurons to build self-organizing structures [23].

For the second challenge, the Broad Learning System network (BLS) [24], [25] can expand in width, so as to provide a faster response scheme with higher accuracy compared with other popular neural networks. BLS is designed based on the random vector functional-link neural network (RVFLNN) [26], which further solves shortcomings of RVLFNN in processing large-scale and high-dimensional data. Also, the computational

This work was supported by the Natural Science Foundation of Fujian Province of China (No. 2021J01002).

speed of a BLS network is faster than an extremely learning machine (ELM) [27]. In addition, Fei et al. proposed a double loop recurrent neural network (DLRNN) [28], which combines the advantages of internal and external feedback. A DLRNN captures both output state information and internal state information simultaneously to own better approximation performance than other regular recurrent neural networks. Therefore, combining BLS and DLRNN is a potential solution to improve a BLS network's dynamic response-ability.

Based on the above considerations, we propose a self-organizing BLS network-based SMC controller, in which the TOPSIS method is used to build a self-organizing structure to efficiently take advantage of computational resources, and a double loop mechanism inspired by DLRNN is created to improve the BLS network's dynamic characteristics. Two dynamic thresholds: one threshold determines whether to retain or remove the existing feature neurons and the other generates new feature neurons, which are automatically determined to construct the self-organizing structure. The double-loop structure is introduced into the self-organizing BLS, so as to process both internal and output state information simultaneously. The Lyapunov stability theorem is used to guarantee the stability of the proposed controller and derive the updated rules of the parameters in the proposed BLS network. The control performance of the control system is demonstrated through a control simulation of a nonlinear chaotic system and a three-link robot manipulator.

The main contributions of the proposed controller include:

1) The TOPSIS method is used to evaluate outputs of feature neurons and two dynamic thresholds are automatically determined to construct a self-organizing BLS, which meets the requirements of different control states and saves computational resources.
2) A double-loop structure is introduced into the self-organizing BLS: the internal feedback neural network is added to the enhancement neurons to capture the internal state information and the external feedback neural network is added to the output neurons to capture the output state information, so as to improve the dynamic characteristics of self-organizing BLS.

II. Problem formulation

A class of nth-order multi-input multi-output (MIMO) uncertain nonlinear system is described as:

$$x^{(n)}(t) = f\left(\underline{x}(t)\right) + g\left(\underline{x}(t)\right)u(t) + d(t) \tag{1}$$

where $\underline{x}(t) = \left[x^{(n-1)}(t), \ldots, \dot{x}(t), x(t)\right] \in \mathbb{R}^{m\times n}$ is the system state vector, $u = [u_1, u_2, \ldots, u_m]^T \in \mathbb{R}^m$ is the control input vector. Thus, Eq. (1) is rewritten as:

$$\begin{aligned} x^{(n)}(t) &= f_0\left(\underline{x}(t)\right) + \varDelta f\left(\underline{x}(t)\right) \\ &\quad + \left(g_0 + \varDelta g\left(\underline{x}(t)\right)\right)u(t) + d(t) \\ &= f_0\left(\underline{x}(t)\right) + g_0 u(t) + \varepsilon\left(\underline{x}(t), t\right) \end{aligned} \tag{2}$$

where $\varDelta f\left(\underline{x}(t)\right)$ and $\varDelta g\left(\underline{x}(t)\right)$ are the respective uncertain terms in $f\left(\underline{x}(t)\right)$ and $g\left(\underline{x}(t)\right)$, and $\varepsilon\left(\underline{x}(t), t\right) = \varDelta f\left(\underline{x}(t)\right) + \varDelta g\left(\underline{x}(t)\right)u(t) + d(t)$ is the lumped uncertainties and external disturbances. If the lumped uncertainties and external disturbances are ignored, Eq. (2) can be rewritten as:

$$x^{(n)}(t) = f_0\left(\underline{x}(t)\right) + g_0 u(t) \tag{3}$$

where $f_0\left(\underline{x}(t)\right) \in \mathbb{R}^m$ and $g_0 = diag(g_{01}, g_{02}, \ldots, g_{0m}) \in \mathbb{R}^{m\times m}$ are the respective nominal portions of $f\left(\underline{x}(t)\right)$ and $g\left(\underline{x}(t)\right)$. If $\underline{x}_d(t) = \left[x_d^{(n-1)T}(t), \ldots, \dot{x}_d^T(t), x_d^T(t)\right]^T \in \mathbb{R}^{m\times n}$ is the desired signal, then the tracking error vector is defined as:

$$\underline{e}(t) = \left[e^{(n-1)}(t),\ e^{(n-2)}(t), \ldots, \dot{e}(t),\ e(t)\right]^T \in \mathbb{R}^{m\cdot n} \tag{4}$$

where $e(t) = x_d(t) - x(t)$.

Thus, an ideal sliding surface is defined as:

$$\begin{aligned} \underline{\underline{e}}(t) &= \begin{pmatrix} \underline{\underline{e}}_1 \\ \underline{\underline{e}}_2 \\ \vdots \\ \underline{\underline{e}}_m \end{pmatrix} \\ &= \begin{bmatrix} e_1^{(n-1)}(t) + \lambda_{11}e_1^{(n-2)}(t) + \cdots + \lambda_{n1}\int_0^T e_1(t)\,dt \\ e_2^{(n-1)}(t) + \lambda_{12}e_2^{(n-2)}(t) + \cdots + \lambda_{n2}\int_0^T e_2(t)\,dt \\ \vdots \\ e_m^{(n-1)}(t) + \lambda_{1m}e_m^{(n-2)}(t) + \cdots + \lambda_{nm}\int_0^T e_m(t)\,dt \end{bmatrix} \\ &= \begin{bmatrix} 1 & & \lambda_{11} & & \lambda_{n1} & \\ & \ddots & & \ddots & & \ddots \\ & & 1 & & \lambda_{1m} & & \lambda_{nm} \end{bmatrix} \begin{bmatrix} \underline{e}(t) \\ \int_0^T e(t)\,dt \end{bmatrix} \\ &= \overline{J}\begin{bmatrix} \underline{e}(t) \\ \int_0^T e(t)\,dt \end{bmatrix} \end{aligned} \tag{5}$$

where $\overline{J} = [I, J] = [I, \lambda_2 I, \ldots, \lambda_n I] \in \mathbb{R}^{m\times(m+1)n}$, $\lambda_j = [\lambda_{1j}, \lambda_{2j}, \ldots, \lambda_{nj}]^T \in \mathbb{R}^n, j = 1, 2, \ldots, m.$

Take the derivative of Eq. (5), we have:

$$\begin{aligned} \dot{\underline{\underline{e}}}(t) &= \overline{J}\begin{bmatrix} \dot{\underline{e}}(t) \\ e(t) \end{bmatrix} = \overline{J}\begin{bmatrix} e^{(n)}(t) \\ \underline{e}(t) \end{bmatrix} \\ &= e^{(n)}(t) + J\underline{e}(t) = x_d^{(n)}(t) - x^{(n)}(t) + J\underline{e}(t) \\ &= x_d^{(n)}(t) - f_0\left(\underline{x}(t)\right) - g_0 u(t) - \varepsilon\left(\underline{x}(t), t\right) + J\underline{e}(t) \end{aligned} \tag{6}$$

If the following inequality (7) can be satisfied, the control system will be stable; meanwhile, an ideal control output u_{IDEAL} can be obtained.

$$\frac{1}{2}\frac{d}{dt}\left(\underline{\underline{e}}_i^2\right) \le -\sum_{i=1}^{m}\sigma_i\left|\underline{\underline{e}}_i\right| \tag{7}$$

where $\sigma_i > 0, i = 1, 2, \ldots, m.$

Applying Eq. (6) into Eq. (7):

$$\begin{aligned} \underline{\underline{e}}^T(t)\,\dot{\underline{\underline{e}}}(t) &= \underline{\underline{e}}^T(t)\left[x_d^{(n)}(t) - f_0\left(\underline{x}(t)\right) - g_0 u(t)\right. \\ &\quad \left. -\varepsilon\left(\underline{x}(t), t\right) + J\underline{e}(t)\right] \le -\sum_{i=1}^{m}\sigma_i\left|\underline{\underline{e}}_i\right| \end{aligned} \tag{8}$$

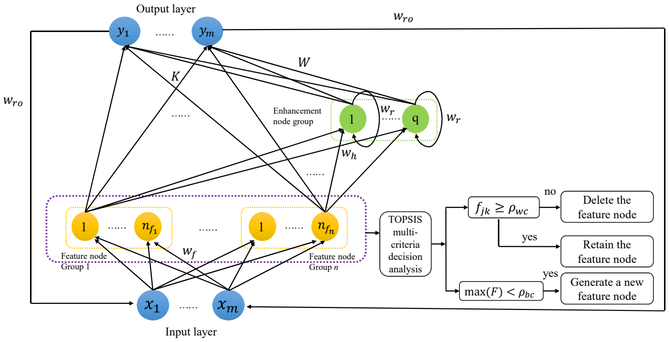


Fig. 1. Structure of proposed SODLRBLS network.

$$
\begin{aligned}
u_{IDEAL} = g_0^{-1}\Big[& x_d^{(n)}(t) - f_0\left(\underline{x}(t)\right) \\
& -\varepsilon\left(\underline{x}(t), t\right) + J\underline{e}(t) + \sigma sgn\left(\underline{\underline{e}}(t)\right)\Big]
\end{aligned} \tag{9}
$$

where $sgn(\cdot)$ is a sign function.

However, the ideal control output u_{IDEAL} cannot be obtained directly because $\varepsilon\left(\underline{x}(t), t\right)$ is unknown, and in general we must know the exact system dynamics parameters to obtain the functional form of f_0 and g_0.

In this paper, TOPSIS-based self-organizing double loop recurrent broad learning system (SODLRBLS) is proposed to fit this process. The input of SODLRBLS is the combination error $\underline{\underline{e}}(t)$, and its output is $u_{SODLRBLS}$.

III. Self-Organizing Double Loop Recurrent Broad Learning System

The TOPSIS method and DLRNN structure are introduced into BLS to improve the performance of the control system in this paper. TOPSIS method is applied to evaluate the output of the feature node layer of BLS to automatically determine two dynamic thresholds for constructing a self-organizing BLS. the internal feedback neural network of DLRNN is added to the enhancement node layer and the external feedback neural network of DLRNN is added to the output layer. The configuration of the proposed SODLRBLS network is illustrated in Fig. 1. These layers of SODLRBLS are specified as below.

1. **Input layer**: both input signals and external feedback signals from output layer are input to every node in this layer. The output of ith node in this layer can be denoted as:

$$
\theta_i = \frac{x_i w_{roi} exu_i}{eexu_i}, i = 1, 2, \ldots, m \tag{10}
$$

where x_i is the input signal to ith node of this layer; exu_i is the ith output signal of SODLRBLS calculated in the last step and $eexu_i$ is the output signal one step before exu_i; exu_i and $eexu_i$ work as the feedback signals in the external recurrent loop; w_{roi} are the neural weights connecting the ith node of the output layer and the ith node of the input layer.

2. **Feature node layer**: there are m feature node groups, each of which has n_{f_j} nodes. w_{fijk} is the weight connecting the ith input node in input layer and the kth feature node in jth feature node group, b_{fjk} is the bias term of the kth feature node in jth feature node group. The output of the kth feature node in jth feature node group is defined by:

$$
\begin{aligned}
f_{jk} &= \sum_{i=1}^{m} \emptyset(w_{fijk}\theta_i + b_{fjk}) \\
&= \sum_{i=1}^{m} \emptyset(w_{fijk}\frac{x_i w_{roi} exu_i}{eexu_i} + b_{fjk})
\end{aligned} \tag{11}
$$

where $\emptyset(\cdot)$ is the activation function, $\emptyset(x) = \tanh(x) = (e^x - e^{-x})/\left(e^x + e^{-x}\right)$.

We denote $F = \left[f_{11}, \ldots, f_{1n_f}, \ldots, f_{m1}, \ldots, f_{mn_f}\right]^T \in \mathbb{R}^{mn_f}$, $n_f = max_{j=1}^{m}(n_{f_j})$, which indicates n_f equals the maximum number of nodes in all feature node groups. For the convenience of calculation, if the feature node group whose number of feature nodes is less than n_f, the output value f is regarded as 0.

The TOPSIS method is used to determine the dynamic deleting threshold value and dynamic generating threshold value, which are used for automatically retaining, deleting the feature node, or generating a new feature node. The Shannon entropy method is used to determine the weights for the evaluation criteria in the TOPSIS. The specific process based on [29], [30] is as follows:

Let feature nodes be the alternatives and let the outputs $f_{j1}, f_{j2}, \ldots, f_{jn_{f_j}}$ be judgment conditions, we have:

1) Generate an evaluation vector:

$$
\boldsymbol{f} = [f_{j1}, f_{j2}, \ldots, f_{jn_f}] \tag{12}
$$

which means each feature node group is an evaluation vector and the output of the nodes in the group constitutes the dimensions of the evaluation vector. There are a total of n evaluation vectors, each evaluation vector has n_f dimensions. Then we have: $\boldsymbol{Eva} = [\boldsymbol{f_1}, \boldsymbol{f_2}, \ldots, \boldsymbol{f_n}]^T$.

2) Normalize evaluation vector to limit the value of each element to $[0, 1]$ by:

$$
\begin{aligned}
b_{ij} &= \frac{\boldsymbol{Eva}_{ij}}{\sqrt{\sum_{i=1}^{n} \boldsymbol{Eva}_{ij}}}, \\
i &= 1, 2, \ldots, n, \; j = 1, 2, \ldots, n_f
\end{aligned} \tag{13}
$$

3) Use the entropy weight method to obtain the weights. The entropy method is given by [31], [32]. First, the entropy value is calculated by:

$$
E_j = -\frac{1}{\ln(n)} \sum_{i=1}^{n} \left(b_{ij} \ln\left(b_{ij}\right)\right) \tag{14}
$$

Then, determine the diversification degree of the measurement quality by:

$$
D_j = 1 - E_j \tag{15}
$$

Determine the weight for each evaluation criteria by:

$$
\varphi_j = \frac{D_j}{\sum_{j=1}^{n_f} D_j} \tag{16}
$$

4) Determine the weighted normalized decision matrix by:

$$v_{ij} = b_{ij}\varphi_j, i = 1, 2, \ldots, n, j = 1, 2, \ldots, n_f \tag{17}$$

5) Determine the best weighted vector $\boldsymbol{v}^*$ and worst vector $\boldsymbol{v}'$ by:

$$v_j^* = \max_{i=1}^{n}(v_{ij}) \tag{18}$$

$$v_j' = \min_{i=1}^{n}(v_{ij}) \tag{19}$$

6) Calculate the separation distance between the best weighted vector $\boldsymbol{v}^*$ and the worst weighted vector $\boldsymbol{v}'$ by:

$$S_i^* = \sqrt{\sum_{j=1}^{n_f} \left(v_j^* - v_{ij}\right)^2} \tag{20}$$

$$S_i' = \sqrt{\sum_{j=1}^{n_f} \left(v_j' - v_{ij}\right)^2} \tag{21}$$

7) Calculate the similarity to the worst condition and the best condition by:

$$wc_i = \frac{S_i^*}{S_i^* + \ S_i'} \tag{22}$$

$$bc_j = \frac{S_i'}{S_i^* + \ S_i'} \tag{23}$$

Vector $\boldsymbol{wc}$ and $\boldsymbol{bc}$ can be used to determine the dynamic thresholds.

$$\rho_{wc} = P_{wc} \min\left(\boldsymbol{WC}\right) = P_{wc} \min(\max\left(\boldsymbol{wc}\right)) \tag{24}$$

$$\rho_{bc} = P_{bc} \max\left(\boldsymbol{BC}\right) = P_{bc} \max(\min\left(\boldsymbol{bc}\right)) \tag{25}$$

where ρ_{wc} is the dynamic deleting threshold and ρ_{bc} is the dynamic generating threshold. $\boldsymbol{WC}$ is a storage vector that contains the maximum values of $\boldsymbol{wc}$ in each train round and $\boldsymbol{BC}$ is a storage vector that contains the minimum values of $\boldsymbol{bc}$ in each train round. $0 \le P_{wc} \le 1$ is deleting threshold coefficient and $0 \le P_{bc} \le 1$ is the generating threshold coefficient.

With dynamic deleting and dynamic generating thresholds, we can achieve the self-organization in the feature node layer by :

$$f_{jk} = \begin{cases} f_{jk} & \text{if } f_{jk} \ge \rho_{wc}, \\ \phi & \text{if } f_{jk} < \rho_{wc}. \end{cases} \tag{26}$$

If the output of the feature node is greater than or equal to ρ_{wc}, this node remains, else this node is deleted. Also, If $max(F) < \rho_{bc}$, a new feature node is generated; otherwise, no node is added.

3. **Enhancement node layer**: Assume that there is one node group with n_h nodes, thus both signals from the feature node layer and internal feedback signals are input to each node in this layer. The output of qth node in this layer is defined as:

$$h_q = e^{-net_q},$$

$$net_q = \sum_{j=1}^{m} \sum_{k=1}^{n_{f_j}} \frac{\left(w_{hjkq} f_{jk} + w_{rq} exh_q - c_{jkq}\right)^2}{v_{jkq}^2} \tag{27}$$

where w_{hjkq} denotes the weights connecting the k-th feature node in j-th feature node group and the q-th enhancement node in this layer; the Gaussian function is the activation function; w_r is the weights of internal feedback; exh is the output of enhancement layer calculated in the last epoch, which works as the feedback signal in the internal recurrent loop. c_{jkq} and v_{jkq} are the center and width of the Gaussian function, which calculated in the q-th enhancement node for the k-th feature node in j-th feature node group. Thus, we have:

$$H = [h_1, h_2, \ldots, h_{n_h}]^T \in \mathbb{R}^{n_h}. \tag{28}$$

4. **Output layer**: The outputs of both the feature node layer and enhancement node layer are passed to this layer. k_{ijk} is the weight connecting kth feature node in jth feature node group and the ith output node, w_{iq} is the weight connecting qth enhancement node and the ith output node. The output of ith output node can be calculated by:

$$u_{SODLRBLS_i} = \sum_{j=1}^{m} \sum_{k=1}^{n_f} k_{ijk} f_{jk} + \sum_{q=1}^{n_h} w_{iq} h_q. \tag{29}$$

We define K and W as:

$$\begin{aligned} K &= [k_{1jk}, k_{2jk}, \ldots, k_{mjk}] \\ &= \begin{bmatrix} k_{111} & k_{211} & \cdots & k_{m11} \\ \vdots & \vdots & & \vdots \\ k_{11n_f} & k_{21n_f} & \cdots & k_{m1n_f} \\ k_{121} & k_{221} & \cdots & k_{m21} \\ \vdots & \vdots & & \vdots \\ k_{12n_f} & k_{22n_f} & \cdots & k_{m2n_f} \\ \vdots & \vdots & & \vdots \\ k_{1m1} & k_{2m1} & \cdots & k_{mm1} \\ \vdots & \vdots & & \vdots \\ k_{1mn_f} & k_{2mn_f} & \cdots & k_{mmn_f} \end{bmatrix} \in \mathbb{R}^{mn_f \times m} \end{aligned} \tag{30}$$

$$\begin{aligned} W &= [w_{1q}, w_{2q}, \ldots, w_{mq}] \\ &= \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{m1} \\ w_{12} & w_{22} & \cdots & w_{m2} \\ \vdots & \vdots & & \vdots \\ w_{1n_h} & w_{2n_h} & \cdots & w_{mn_h} \end{bmatrix} \in \mathbb{R}^{n_h \times m}. \end{aligned} \tag{31}$$

Thus, we have:

$$u_{SODLRBLS} = K^T F + W^T H \tag{32}$$

The overall computational process of the proposed SODLRBLS network is summarized in pseudocode in Algorithm 1.

IV. Adaptive Learning Algorithm and Convergence Analysis

The framework of the proposed SODLRBLS-based control system is illustrated in Fig. 2. The SODLRBLS is used as the main controller in the control system, which works with a robust controller.

Algorithm 1 The pseudocode of SODLRBLS network.

```
calculate $\theta_i$ using Eq. (10);
calculate $f_{jk}$ using Eq. (11);
calculate dynamic deleting $\rho_{wc}$ and generating threshold $\rho_{bc}$ using Eqs. (24) and (25);
Self-organizing of enhancement node layer according to self-organization rule (26);
calculate $h_q$ using Eq. (27);
calculate the output $u$ of network using Eq. (29);
update $\dot{\hat{K}}, \dot{\hat{W}}, \dot{\widehat{w_f}}, \dot{\widehat{b_f}}, \dot{\widehat{w_{ro}}}, \dot{\widehat{w_h}}, \dot{\hat{c}}, \dot{\hat{v}}$ and $\dot{\widehat{w_r}}$ using updating rules (46)-(54).
```

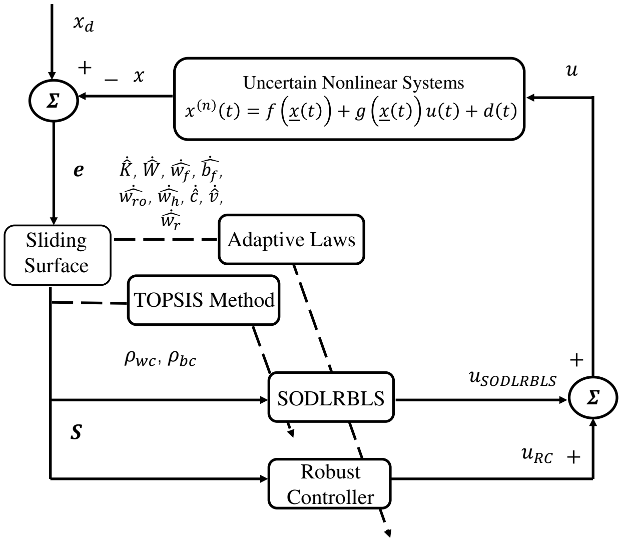

x_d
+
−
x
Σ
Uncertain Nonlinear Systems
$x^{(n)}(t) = f\left(\underline{x}(t)\right) + g\left(\underline{x}(t)\right)u(t) + d(t)$
u
e
$\dot{\hat{K}}, \dot{\hat{W}}, \dot{\widehat{w_f}}, \dot{\widehat{b_f}}, \dot{\widehat{w_{ro}}}, \dot{\widehat{w_h}}, \dot{\hat{c}}, \dot{\hat{v}}, \dot{\widehat{w_r}}$
Sliding Surface
Adaptive Laws
TOPSIS Method
ρ_{wc}, ρ_{bc}
SODLRBLS
$u_{SODLRBLS}$
+
Σ
S
u_{RC}
+
Robust Controller

Fig. 2. The proposed SODLRBLS-based Control System.

A set of update laws for SODLRBLS are derived to support the proposed control system and the rules can be proven that global control system can achieve an H^∞ tracking control effect using the Lyapunov stability theory. The detailed proof process is as follows:

Subtracting Eq. (9) from Eq. (6)

$$\dot{\underline{e}}(t) = g_0\left[u_{IDEAL} - u\right] - \sigma sgn(\underline{e}(t)) \tag{33}$$

Assume that there exists an optimal SODLRBLS, $u^*_{SODLRBLS}$, to approach an ideal controller u_{IDEAL}. K^* and W^* are optimal weight matrices, and F^* and H^* are optimal output matrices of the feature node layer and the enhancement node layer in the optimal SODLRBLS. Then the optimal control output is:

$$\begin{aligned} u_{IDEAL} = u^*_{SODLRBLS} + \varepsilon &= \left(K^T F + W^T H\right)^* + \varepsilon \\ &= K^{*T} F^* + W^{*T} H^* + \varepsilon \end{aligned} \tag{34}$$

where ε is a minimum approximation error vector; $\hat{K}$ and $\hat{W}$ are estimated weight matrices; and $\hat{F}$ and $\hat{H}$ are estimated output matrices of the feature node layer and enhancement node layer in the actual SODLRBLS. Then the actual control output is:

$$u = u_{SODLRBLS} + u_{RC} = \hat{K}^T \hat{F} + \hat{W}^T \hat{H} + u_{RC} \tag{35}$$

where u_{RC} is the output of the robust controller.

Taking Eqs. (34) and (35) into Eq. (33):

$$\begin{aligned} \dot{\underline{e}}(t) &= g_0\Big[K^{*T} F^* + W^{*T} H^* + \varepsilon - \hat{K}^T \hat{F} - \hat{W}^T \hat{H} \\ &\qquad - u_{RC}\Big] - \sigma sgn\left(\underline{e}(t)\right) \\ &= g_0\left[\widetilde{K}^T F^* + \hat{K}^T \widetilde{F} + \widetilde{W}^T H^* + \hat{W}^T \widetilde{H} + \varepsilon - u_{RC}\right] \\ &\qquad - \sigma sgn\left(\underline{e}(t)\right) \end{aligned} \tag{36}$$

where $\widetilde{K} = K^* - \hat{K}$; $\widetilde{W} = W^* - \hat{W}$; $\widetilde{F} = F^* - \hat{F}$ and $\widetilde{H} = H^* - \hat{H}$. A partially linear form of the receptive-field basis function vector $\widetilde{F}$ in Taylor series can be described as:

$$\begin{aligned} \widetilde{F} = \begin{pmatrix} \widetilde{f_1} \\ \vdots \\ \widetilde{f_{n_d}} \end{pmatrix} &= \begin{pmatrix} \left(\frac{\partial f_1}{\partial w_f}\right)^T \\ \vdots \\ \left(\frac{\partial f_{n_d}}{\partial w_f}\right)^T \end{pmatrix} |_{w_f=\widehat{w_f}} \left({w_f}^* - \widehat{w_f}\right) \\ &+ \begin{pmatrix} \left(\frac{\partial f_1}{\partial b_f}\right)^T \\ \vdots \\ \left(\frac{\partial f_{n_d}}{\partial b_f}\right)^T \end{pmatrix} |_{b_f=\widehat{b_f}} \left({b_f}^* - \widehat{b_f}\right) \\ &+ \begin{pmatrix} \left(\frac{\partial f_1}{\partial w_{ro}}\right)^T \\ \vdots \\ \left(\frac{\partial f_{n_d}}{\partial w_{ro}}\right)^T \end{pmatrix} |_{w_{ro}=\widehat{w_{ro}}} \left({w_{ro}}^* - \widehat{w_{ro}}\right) + \beta_1 \\ &= F_{w_f}\widetilde{w_f} + F_{b_f}\widetilde{b_f} + F_{w_{ro}}\widetilde{w_{ro}} + \beta_1 \end{aligned} \tag{37}$$

where F_{w_f}, F_{b_f} and $F_{w_{ro}}$ are defined by:

$$\begin{cases} F_{w_f} = \left[\frac{\partial f_1}{\partial w_f}, \ldots, \frac{\partial f_{n_d}}{\partial w_f}\right]^T |_{w_f=\widehat{w_f}} \in \mathbb{R}^{n_d \times n_f n_d} \\ F_{b_f} = \left[\frac{\partial f_1}{\partial b_f}, \ldots, \frac{\partial f_{n_d}}{\partial b_f}\right]^T |_{b_f=\widehat{b_f}} \in \mathbb{R}^{n_d \times n_f n_d} \\ F_{w_{ro}} = \left[\frac{\partial f_1}{\partial w_{ro}}, \ldots, \frac{\partial f_{n_d}}{\partial w_{ro}}\right]^T |_{w_{ro}=\widehat{w_{ro}}} \in \mathbb{R}^{n_d \times n_f n_d} \end{cases} \tag{38}$$

where $\widetilde{w_f} = {w_f}^* - \widehat{w_f}$; $\widetilde{b_f} = {b_f}^* - \widehat{b_f}$, $\widetilde{w_{ro}} = {w_{ro}}^* - \widehat{w_{ro}}$; and β_1 is a higher-order vector.

F^* can also be rewritten as:

$$F^* = \hat{F} + \widetilde{F} = \hat{F} + F_{w_f}\widetilde{w_f} + F_{b_f}\widetilde{b_f} + F_{w_{ro}}\widetilde{w_{ro}} + \beta_1. \tag{39}$$

Also, a partially linear form of the receptive-field basis function vector $\widetilde{H}$ in Taylor series can be described as:

$$\begin{aligned}\widetilde{H} &= \begin{pmatrix} \widetilde{h}_1 \\ \vdots \\ \widetilde{h}_{n_h} \end{pmatrix} = \begin{pmatrix} \left(\frac{\partial h_1}{\partial w_h}\right)^T \\ \vdots \\ \left(\frac{\partial h_{n_h}}{\partial w_h}\right)^T \end{pmatrix} |_{w_h=\widehat{w_h}} \left({w_h}^* - \widehat{w_h}\right) \\ &+ \begin{pmatrix} \left(\frac{\partial h_1}{\partial c}\right)^T \\ \vdots \\ \left(\frac{\partial h_{n_h}}{\partial c}\right)^T \end{pmatrix} |_{c=\hat{c}} \left(c^* - \hat{c}\right) + \begin{pmatrix} \left(\frac{\partial h_1}{\partial v}\right)^T \\ \vdots \\ \left(\frac{\partial h_{n_h}}{\partial v}\right)^T \end{pmatrix} |_{v=\hat{v}} \left(v^* - \hat{v}\right) \\ &+ \begin{pmatrix} \left(\frac{\partial h_1}{\partial w_r}\right)^T \\ \vdots \\ \left(\frac{\partial h_{n_h}}{\partial w_r}\right)^T \end{pmatrix} |_{w_r=\widehat{w_r}} \left({w_r}^* - \widehat{w_r}\right) + \beta_2 \\ &= H_{w_h}\widetilde{w_h} + H_c\widetilde{c} + H_v\widetilde{v} + H_{w_r}\widetilde{w_r} + \beta_2 \end{aligned} \tag{40}$$

where H_{w_h}, H_c, H_v and H_{w_r} are defined by:

$$\begin{cases} H_{w_h} = \left[\frac{\partial h_1}{\partial w_h}, \ldots, \frac{\partial h_{n_h}}{\partial w_h}\right]^T |_{w_h=\widehat{w_h}} \in \mathbb{R}^{n_h \times mn_h} \\ H_c = \left[\frac{\partial h_1}{\partial c}, \ldots, \frac{\partial h_{n_h}}{\partial c}\right]^T |_{c=\hat{c}} \in \mathbb{R}^{n_h \times mn_h} \\ H_v = \left[\frac{\partial h_1}{\partial v}, \ldots, \frac{\partial h_{n_h}}{\partial v}\right]^T |_{v=\hat{v}} \in \mathbb{R}^{n_h \times mn_h} \\ H_{w_r} = \left[\frac{\partial h_1}{\partial w_r}, \ldots, \frac{\partial h_{n_h}}{\partial w_r}\right]^T |_{w_r=\widehat{w_r}} \in \mathbb{R}^{n_h \times mn_h} \end{cases} \tag{41}$$

where $\widetilde{w_h} = w_h^* - \hat{w}_h$, $\widetilde{c} = c^* - \hat{c}$, $\widetilde{v} = v^* - \hat{v}$, $\widetilde{w_r} = {w_r}^* - \widehat{w_r}$, and β_2 is a higher-order vector.

H^* can also be rewritten as:

$$H^* = \hat{H} + \widetilde{H} = \hat{H} + H_{w_h}\widetilde{w_h} + H_c\widetilde{c} + H_v\widetilde{v} + H_{w_r}\widetilde{w_r} + \beta_2. \tag{42}$$

Then, substituting Eqs. (39) and (42) to Eq. (36), we have:

$$\begin{aligned}\dot{\underline{\underline{e}}}(t) &= g_0 \Big[\widetilde{K}^T \left(\hat{F} + F_{w_f}\widetilde{w_f} + F_{b_f}\widetilde{b_f} + F_{w_{ro}}\widetilde{w_{ro}} + \beta_1\right) \\ &-\sigma sgn\left(\underline{\underline{e}}(t)\right) + \hat{K}^T \left(F_{w_f}\widetilde{w_f} + F_{b_f}\widetilde{b_f} + F_{w_{ro}}\widetilde{w_{ro}} + \beta_1\right) \\ &-\widetilde{W}^T \left(\hat{H} + H_{w_h}\widetilde{w_h} + H_c\widetilde{c} + H_v\widetilde{v} + H_{w_r}\widetilde{w_r} + \beta_2\right) \\ &-\hat{W}^T \left(H_{w_h}\widetilde{w_h} + H_{{}_c}\widetilde{c} + H_v\widetilde{v} + H_{w_r}\widetilde{w_r} + \beta_2\right) + \varepsilon - u_{RC}\Big] \\ &= g_0 \Big[\hat{K}^T \left(F_{w_f}\widetilde{w_f} + F_{b_f}\widetilde{b_f} + F_{w_{ro}}\widetilde{w_{ro}}\right) \\ &-\hat{W}^T \left(H_{w_h}\widetilde{w_h} + H_{{}_c}\widetilde{c} + H_v\widetilde{v} + H_{w_r}\widetilde{w_r}\right) \\ &+\widetilde{K}^T\hat{F} - \widetilde{W}^T\hat{H} + \tau - u_{RC}\Big] - \sigma sgn\left(\underline{\underline{e}}(t)\right) \end{aligned} \tag{43}$$

where $\tau = K^{*T}\beta_1 + W^{*T}\beta_2$
$+\widetilde{K}^T \left(F_{w_f}\widetilde{w_f} + F_{b_f}\widetilde{b_f} + F_{w_{ro}}\widetilde{w_{ro}}\right)$
$+\widetilde{W}^T \left(H_{w_h}\widetilde{w_h} + H_{{}_c}\widetilde{c} + H_v\widetilde{v} + H_{w_r}\widetilde{w_r}\right) + \varepsilon$ is a combined error of SODLRBLS.

Because of the existence of τ, we can use an attenuation constant, λ, to guarantee an H^∞ tracking performance [33]:

$$\begin{aligned}&\sum_{i=1}^{m}\int_0^T \underline{\underline{e}}_i^2(t)dt \le \underline{\underline{e}}^T(0)\, g_0^{-1}\underline{\underline{e}}(0) \\ &+ tr\left[\widetilde{K}^T(0)\,\eta_K^{-1}\widetilde{K}(0)\right] + tr\left[\widetilde{W}^T(0)\,\eta_W^{-1}\widetilde{W}(0)\right] \\ &+ \widetilde{w_f}^T(0)\,\eta_{w_f}^{-1}\widetilde{w_f}(0) + \widetilde{b_f}^T(0)\,\eta_{b_f}^{-1}\widetilde{b_f}(0) \\ &+ \widetilde{w_{ro}}^T(0)\,\eta_{w_{ro}}^{-1}\widetilde{w_{ro}}(0) + \widetilde{w_h}^T(0)\,\eta_{w_h}^{-1}\widetilde{w_h}(0) \\ &+ \widetilde{c}^T(0)\,\eta_c^{-1}\widetilde{c}(0) + \widetilde{v}^T(0)\,\eta_v^{-1}\widetilde{v}(0) \\ &+ \widetilde{w_r}^T(0)\,\eta_{w_r}^{-1}\widetilde{w_r}(0) + \sum_{i=1}^{m}\lambda_i^2\int_0^T \tau_i^2(t)dt \end{aligned} \tag{44}$$

where η_K, η_W, η_{w_f}, η_{b_f}, $\eta_{w_{ro}}$, η_{w_h}, η_c, η_v and η_{w_r} are diagonal positive constant learning-rate matrices. The initial conditions of system are set as $\underline{\underline{e}}(0) = 0$,$\widetilde{K}(0) = 0$, $\widetilde{W}(0) = 0$, $\widetilde{w_f}(0) = 0$, $\widetilde{b_f}(0) = 0$, $\widetilde{w_{ro}}(0) = 0$, $\widetilde{w_h}(0) = 0$, $\widetilde{c}(0) = 0$, $\widetilde{v}(0) = 0$, $\widetilde{w_r}(0) = 0$, so Eq. (44) can be rewritten as:

$$\sum_{i=1}^{m}\int_0^T \underline{\underline{e}}_i^2(t)dt \le \sum_{i=1}^{m}\lambda_i^2\int_0^T \tau_i^2(t)dt \tag{45}$$

The update laws of all parameters in SODLRBLS-based controller proposed in this paper is described as:

$$\dot{\hat{K}} = \eta_K\hat{F}\underline{\underline{e}}^T(t) \tag{46}$$

$$\dot{\hat{W}} = \eta_W\hat{H}\underline{\underline{e}}^T(t) \tag{47}$$

$$\dot{\widehat{w_f}} = \eta_{w_f}F_{w_f}^T\hat{K}\underline{\underline{e}}^T(t) \tag{48}$$

$$\dot{\widehat{b_f}} = \eta_{b_f}F_{b_f}^T\hat{K}\underline{\underline{e}}^T(t) \tag{49}$$

$$\dot{\widehat{w_{ro}}} = \eta_{w_{ro}}F_{w_{ro}}^T\hat{K}\underline{\underline{e}}^T(t) \tag{50}$$

$$\dot{\widehat{w_h}} = \eta_{w_h}H_{w_h}^T\hat{W}\underline{\underline{e}}^T(t) \tag{51}$$

$$\dot{\hat{c}} = \eta_c H_c^T\hat{W}\underline{\underline{e}}^T(t) \tag{52}$$

$$\dot{\hat{v}} = \eta_v H_v^T\hat{W}\underline{\underline{e}}^T(t) \tag{53}$$

$$\dot{\widehat{w_r}} = \eta_{w_r}H_{w_r}^T\hat{W}\underline{\underline{e}}^T(t) \tag{54}$$

$$u_{RC} = \left(2R^2\right)^{-1}\left[IR^2 + I\right]\underline{\underline{e}}^T(t) \tag{55}$$

where Eq. (55) is the adaptive laws of robust controller, $R = diag[\lambda_1\, \lambda_2, \ldots, \lambda_m] \in \mathbb{R}^{m\times m}$ is a diagonal matrix of robust controller, whose elements are used as attenuation coefficients.

The Lyapunov function of the control system is empirically designed as:

$$\begin{aligned}&L\left(\underline{\underline{e}}(t), \widetilde{K}, \widetilde{W}, \widetilde{w_f}, \widetilde{b_f}, \widetilde{w_{ro}}, \widetilde{w_h}, \widetilde{c}, \widetilde{v}, \widetilde{w_r}\right) \\ &= \frac{1}{2}[\underline{\underline{e}}^T g_0^{-1}\underline{\underline{e}} + tr\left[\widetilde{K}^T\eta_W^{-1}\widetilde{K}\right] + tr\left[\widetilde{W}^T\eta_W^{-1}\widetilde{W}\right] \\ &+ \widetilde{w_f}^T\eta_{w_f}^{-1}\widetilde{w_f} + \widetilde{b_f}^T\eta_{b_f}^{-1}\widetilde{b_f} + \widetilde{w_{ro}}^T\eta_{w_{ro}}^{-1}\widetilde{w_{ro}} \\ &+ \widetilde{w_h}^T\eta_{w_h}^{-1}\widetilde{w_h} + \widetilde{c}^T\eta_c^{-1}\widetilde{c} + \widetilde{v}^T\eta_v^{-1}\widetilde{v} + \widetilde{w_r}^T\eta_{w_r}^{-1}\widetilde{w_r}]. \end{aligned} \tag{56}$$

Taking the derivative of the Lyapunov function and using Eq. (36), we have:

$$\begin{aligned}
&\dot{L}\left(\underline{\underline{e}}\left(t\right),\widetilde{K},\widetilde{W},\widetilde{w_f},\widetilde{b_f},\widetilde{w_{ro}},\widetilde{w_h},\widetilde{c},\widetilde{v},\widetilde{w_r}\right)\\
&=\underline{\underline{e}}^T g_0^{-1}\dot{\underline{\underline{e}}}-tr\left[\widetilde{K}^T\eta_K^{-1}\dot{\hat{K}}\right]-tr\left[\widetilde{W}^T\eta_W^{-1}\dot{\hat{W}}\right]\\
&-\widetilde{w_f}^T\eta_{w_f}^{-1}\dot{\widehat{w_f}}-\widetilde{b_f}^T\eta_{b_f}^{-1}\dot{\widehat{b_f}}-\widetilde{w_{ro}}^T\eta_{w_{ro}}^{-1}\dot{\widehat{w_{ro}}}\\
&-\widetilde{w_h}^T\eta_{w_h}^{-1}\dot{\widehat{w_h}}-\widetilde{c}^T\eta_c^{-1}\dot{\hat{c}}-\widetilde{v}^T\eta_v^{-1}\dot{\hat{v}}-\widetilde{w_r}^T\eta_{w_r}^{-1}\dot{\widehat{w_r}}\\
&\leq -tr\left[\widetilde{K}^T\left(\eta_K^{-1}\dot{\hat{K}}-\hat{F}\underline{\underline{e}}^T\right)\right]-tr\left[\widetilde{W}^T\left(\eta_W^{-1}\dot{\hat{W}}-\hat{H}\underline{\underline{e}}^T\right)\right]\\
&+\widetilde{w_f}^T\left[\hat{K}F_{w_f}\underline{\underline{e}}^T-\eta_{w_f}^{-1}\dot{\widehat{w_f}}\right]+\widetilde{b_f}^T\left[\hat{K}F_{b_f}\underline{\underline{e}}^T-\eta_{b_f}^{-1}\dot{\widehat{b_f}}\right]\\
&+\widetilde{w_{ro}}^T\left[\hat{K}F_{w_{ro}}\underline{\underline{e}}^T-\eta_{w_{ro}}^{-1}\dot{\widehat{w_{ro}}}\right]+\widetilde{w_h}^T\left[\hat{W}H_{w_h}\underline{\underline{e}}^T-\eta_{w_h}^{-1}\dot{\widehat{w_h}}\right]\\
&+\widetilde{c}^T\left[\hat{W}H_c\underline{\underline{e}}^T-\eta_c^{-1}\dot{\hat{c}}\right]+\widetilde{v}^T\left[\hat{W}H_v\underline{\underline{e}}^T-\eta_v^{-1}\dot{\hat{v}}\right]\\
&+\widetilde{w_r}^T\left[\hat{W}H_{w_r}\underline{\underline{e}}^T-\eta_{w_r}^{-1}\dot{\widehat{w_r}}\right]+\underline{\underline{e}}^T\left(\tau-u_{RC}\right). \qquad (57)
\end{aligned}$$

Substituting from Eqs. (46) to (55) into (57), we have:

$$\begin{aligned}
&\dot{L}\left(\underline{\underline{e}}\left(t\right),\widetilde{K},\widetilde{W},\widetilde{w_f},\widetilde{b_f},\widetilde{w_{ro}},\widetilde{w_h},\widetilde{c},\widetilde{v},\widetilde{w_r}\right)\\
&\leq \underline{\underline{e}}^T\left(\tau-u_{RC}\right)=-\frac{1}{2}\underline{\underline{e}}^T\underline{\underline{e}}-\frac{1}{2}\frac{\underline{\underline{e}}^T\underline{\underline{e}}}{\lambda^2}+\underline{\underline{e}}^T\tau\\
&=-\frac{1}{2}\underline{\underline{e}}^T\underline{\underline{e}}-\frac{1}{2}\left[\frac{\underline{\underline{e}}}{\lambda}-\lambda\tau\right]^T\left[\frac{\underline{\underline{e}}}{\lambda}-\lambda\tau\right]+\frac{1}{2}\lambda^2\tau^T\tau\\
&\leq -\frac{1}{2}\underline{\underline{e}}^T\underline{\underline{e}}+\frac{1}{2}\lambda^2\tau^T\tau. \qquad (58)
\end{aligned}$$

Integrating Eq. (58) from $t=0$ to $t=T$, we have:

$$L\left(T\right)-L\left(0\right)\leq -\frac{1}{2}\sum_{i=0}^{m}\int_0^T\underline{\underline{e}}_i^2(t)dt+\frac{1}{2}\sum_{i=1}^{m}\lambda_i^2\int_0^T\tau_i^2(t)dt. \qquad (59)$$

Since $L\left(T\right)>0$, we have:

$$\begin{aligned}
&\sum_{i=1}^{m}\int_0^T\underline{\underline{e}}_i^2\left(t\right)dt\leq \underline{\underline{e}}^T\left(0\right)g_0^{-1}\underline{\underline{e}}\left(0\right)\\
&+tr\left[\widetilde{K}^T\left(0\right)\eta_K^{-1}\widetilde{K}\left(0\right)\right]+tr\left[\widetilde{W}^T\left(0\right)\eta_W^{-1}\widetilde{W}\left(0\right)\right]\\
&+\widetilde{w_f}^T\left(0\right)\eta_{w_f}^{-1}\widetilde{w_f}\left(0\right)+\widetilde{b_f}^T\left(0\right)\eta_{b_f}^{-1}\widetilde{b_f}\left(0\right)\\
&+\widetilde{w_{ro}}^T\left(0\right)\eta_{w_{ro}}^{-1}\widetilde{w_{ro}}\left(0\right)+\widetilde{w_h}^T\left(0\right)\eta_{w_h}^{-1}\widetilde{w_h}\left(0\right)\\
&+\widetilde{c}^T\left(0\right)\eta_c^{-1}\widetilde{c}\left(0\right)+\widetilde{v}^T\left(0\right)\eta_v^{-1}\widetilde{v}\left(0\right)\\
&+\widetilde{w_r}^T\left(0\right)\eta_{w_r}^{-1}\widetilde{w_r}\left(0\right)+\sum_{i=1}^{m}\lambda_i^2\int_0^T\tau_i^2(t)dt \qquad (60)
\end{aligned}$$

where Eq. (60) is exactly Eq. (44). The above proves that the tracking control effect of H^∞ can be achieved.

V. Simulation results

In this section, we first conducted an ablation experiment of SODLRBLS to show each component's importance in the Duffing-Holmes chaotic system. Then, a comparison with other neural network-based controllers was conducted in a three-link robotic manipulator to exhibit the advantages of our proposed work.

A. Ablation Experiment: Duffing–Holmes Chaotic System

In the ablation experiment, the backbone of SODLRBLS was denoted as BLS; the self-organizing behavior of SODLRBLS was deleted, denoted as DLRBLS; and the DLRNN structure of SODLRBLS was deleted, denoted as SOBLS. The performances of BLS, DLRBLS, SOBLS, and SODLRBLS were compared to study the influence of self-organizing behavior and DLRNN structure on the performance of the SODLRBLS network. The control object is a Duffing-Holmes chaotic system, whose dynamic equation is:

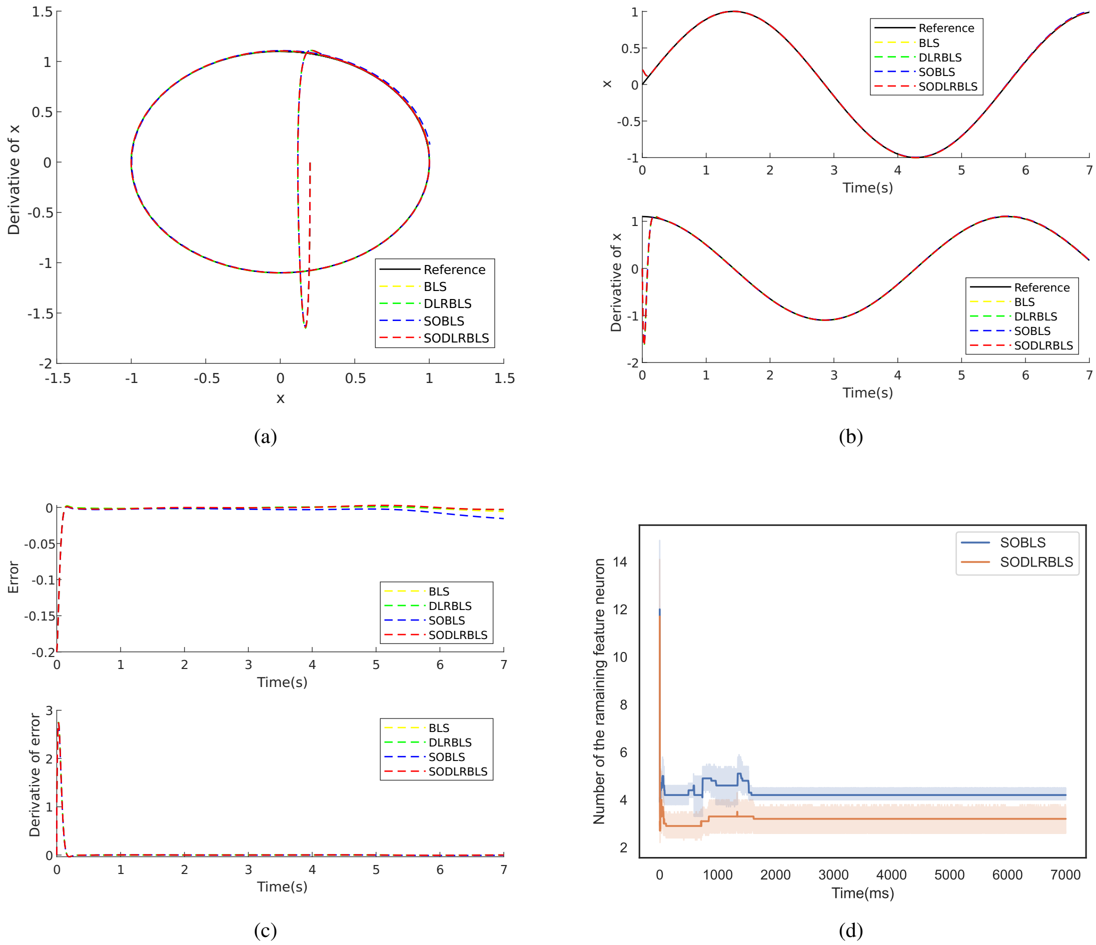
$$\begin{aligned}
\ddot{x}&=f\left(x\right)+u\left(t\right)+f_d\left(x\right)+d\left(t\right)\\
&=-0.25\dot{x}+x-x^3+0.1\sqrt{x^2+\dot{x}^2}\sin\left(t\right)\\
&+0.3\cos\left(t\right)+u\left(t\right)+f_d\left(x\right)+d(t) \qquad (61)
\end{aligned}$$

where $f_d\left(x\right)=0.1x$ is the uncertainty, $d\left(t\right)=\sin\left(2t\right)+\cos\left(2t\right)$ is the disturbance. $x\left(0\right)=\dot{x}\left(0\right)=0.2$ are the initial system states. $x_d\left(0\right)=\sin\left(1.1t\right)$ is the reference trajectory. The sliding hyperplane was designed as $s\left(t\right)=0.5\dot{e}\left(t\right)+5e(t)$; and the diagonal matrix R of the robust controller was set to $0.07I$, where $I\in\mathbb{R}^2$ is a unit matrix.

The weights, bias, and parameters of the Gaussian function of those four networks were initialized by random numbers from $[-1,1]$. The learning rate, η, is set to 0.01. The number of feature nodes in each group is set to 3 for BLS and DLRBLS, and the number of feature nodes in each group of SOBLS and SODLRBLS is initialized by random numbers from $[1,10]$. In the process of self-organization, no more than 10 feature nodes can exist in each group. The number of feature node groups was set to 2, and the number of enhancement nodes is set to 1 for these four networks. The deleting threshold coefficient, P_{wc}, was set to 0.1; and the generating threshold coefficient, P_{bc}, was set to 1. We conducted ten simulation experiments for each of the four networks and averaged all results.

The simulation results are shown in Fig. 3. Fig. 3-(a) shows all of the trajectory phase portraits; Fig. 3-(b) shows all of the tracking responses of x and tracking responses of $\dot{x}$; Fig. 3-(c) shows all of tracking errors and derivative of errors. From Fig. 3(a) to (c), all the four controllers (BLS, DLRBLS, SOBLS and SODLRBLS) can well follow the reference trajectories. Moreover, the number of remaining feature nodes of SOBLS and SODLRBLS are shown in Fig. 3(d). In the figure, both SOBLS-based controller and SODLRBLS-based controller can be suitably self-organized. However, the feature nodes of SOBLS changed more drastically, and the number of feature nodes within the converged term is larger that that of SODLRBLS.

In addition, Table I shows the average RMSE values of ten simulations. The average RMSE of the DLRBLS-based controller is significantly smaller than those of other BLS-based controllers, showing that the double-loop recurrent structure can improve the accuracy of the controller well. Also, the performance of DLRBLS and SODLRBLS was very close, but the result of SOBLS was not as good as SODLRBLS; this result proved with the assistance of the TOPSIS structure,



(a)

(b)

(c)

(d)

Fig. 3. The simulation results for a nonlinear chaotic system. (a) The trajectory phase portrait, (b) The tracking responses of x and tracking responses of $\dot{x}$, (c) Tracking errors and derivative of errors, (d) The number of remaining feature nodes of SOBLS and SODLRBLS.

TABLE I
THE AVERAGE RMSE VALUES AND COMPUTATIONAL TIME OF 10 SIMULATIONS OF DUFFING-HOLMES CHAOTIC SYSTEM.

	BLS	DLRBLS	SOBLS	SODLRBLS
Average RMSE	0.01460	0.01358	0.01555	0.01360
Average time (s)	0.54952	0.66873	1.03437	1.21650

SODLRBLS can use fewer neurons to archive equivalent preferment with DLRBLS.

B. Comparison Experiment: Three-Link Robot Manipulator

The model of a three-link robot manipulator is illustrated in Fig. 4. The dynamic equation of the three-link robot manipulator is expressed as:

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + g(q) = u + \tau_d \tag{62}$$

where $M(q)$ is the inertia matrix, $C(q,\dot{q})$ is the Coriolis/Centripetal matrix, $g(q)$ is the gravity vector, u is the output torque, $\tau_d = 2 \times [0.2\sin(2t), 0.1\cos(2t), 0.1\sin(t)]^T$ is the external disturbance, q is the the joint angle state vector, $\dot{q}$ is the velocity vector, $\ddot{q}$ is the acceleration vector.

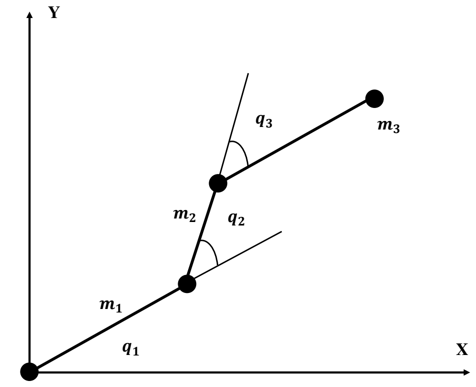


Fig. 4. The model of three-Link Robot Manipulator.

$q = [-0.3, 0.1, -0.4]^T$, and the original state is $\dot{q} = 0$ and $\ddot{q} = 0$. The reference trajectories were set as: $q_{d1} = 0.5 \times [0.5\sin(2t+2.5) + 0.75\cos(2t+1.5),\ \sin(2t) + \sin(t),\ 0.2\cos(2t) - 0.2\sin(t)]^T$, $\dot{q}_{d1} = 0$ and $\ddot{q}_{d1} = 0$. The reference trajectories were changed as $q_{d2} =$

TABLE II
THE AVERAGE RMSE VALUES AND COMPUTATIONAL TIME OF TEN SIMULATIONS OF THREE-LINK ROBOT MANIPULATOR.

	DLRNN	FBEL	SODLRBLS
Joint 1	0.02145	0.01902	**0.01897**
Joint 2	0.01994	0.01845	**0.01795**
Joint 3	0.01925	0.01750	**0.01734**
Average time (s)	**3.59750**	3.69021	13.15782

$0.5 \times [0.5\sin(2t) + \cos(t+1), \sin(2t)\cos(t+1), 0.25 - 0.1\sin(t+1) - 0.1\sin(2t)]^T$, $\dot{q}_{d2} = 0$ and $\ddot{q}_{d2} = 0$ at $15s$ to evaluate the robustness of the proposed control system. The sliding hyperplane was designed as $s(t) = 0.55\dot{e}(t) + 10e(t)$ and the diagonal matrix R of the robust controller was set to $0.066I$, where $I \in \mathbb{R}^2$ is a unit matrix.

The weights, bias, and parameters of the Gaussian function of SODLRBLS are initialized by random numbers from $[-0.01, 0.01]$. The learning rate η is 0.0001. The number of feature nodes in each group of SODLRBLS is initialized by random numbers from $[1, 10]$. The number of feature node groups is set to 3, and the number of enhancement nodes is set to 3. The deleting threshold coefficient P_{wc} was set as 0.1 and The generating threshold coefficient P_{bc} was set as 0.1. We conducted ten simulation experiments for each of the four networks and averaged all results as well.

The simulation results are shown in Fig. 5. Figs. 5(a)-(c) show trajectory responses and tracking errors of Joints 1, 2, and 3. We compared the tracking effect of the SODLRBLS-based controller with the DLRNN-based and FBEL-based controllers. All the controllers can well handle the robotic system; however, for the average RMSE values (given in Table II) of ten simulations of the three-link robot manipulator, the tracking effect of the SODLRBLS-based controller was significantly better than those of the DLRNN-based controller and FBEL-based controllers. In Fig. 5(d), the feature nodes of SODLRBLS can be well self-organized in the initial stage and when the trajectory was abruptly changed, so as to meet the requirements of different control states and save computational resources. Note that the number of the feature nodes contained a chattering phenomenon after 15 seconds, more efforts should be required to eliminate the chattering. Note that we calculated the average computational time of the ten simulations in both Tables I and II. In these two tables, our computational time is larger than those of the compared methods. The reason is the self-organization mechanism increased the amount of computational time.

VI. CONCLUSION

In this work, we proposed a type of self-organizing neural network that is used to build a controller for uncertain nonlinear systems. The network contained the key structure of the TOPSIS method and Broad Learning System. In addition, a double-loop recurrent structure was introduced to improve the network's dynamic characteristics. The Lyapunov stability function was used to prove the stability of the control system and derive the updated rules of the parameters in the proposed network. The network-based control system was used to simulate the control of a Duffing–Holmes chaotic system and a three-link robot manipulator. The simulation results showed that the proposed control system can achieve a better tracking performance against other network-based controllers. The future study will focus on building a more stable organizing method for TOPSIS to reduce the node number's instability. In addition, it is also crucial to apply our system to control real-time models and to deal with the chattering problems in real dynamic systems.

.

REFERENCES

[1] B. Priya and S. S. Manohar, "Adaptive power control and duty cycle based medium access control protocol for cluster based wireless sensor network," *Science and Technology*, vol. 23, no. 1, pp. 38–54, 2020.

[2] B. Luo, Y. Yang, and D. Liu, "Policy iteration q-learning for data-based two-player zero-sum game of linear discrete-time systems," *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3630–3640, 2020.

[3] I. A. Zamfirache, R.-E. Precup, R.-C. Roman, and E. M. Petriu, "Policy iteration reinforcement learning-based control using a grey wolf optimizer algorithm," *Information Sciences*, vol. 585, pp. 162–175, 2022.

[4] Z. Vukic, *Nonlinear control systems*. CRC Press, 2003.

[5] H. Tsukamoto and S.-J. Chung, "Robust controller design for stochastic nonlinear systems via convex optimization," *IEEE Transactions on Automatic Control*, vol. 66, no. 10, pp. 4731–4746, 2020.

[6] I. Al-Darraji, D. Piromalis, A. A. Kakei, F. Q. Khan, M. Stojmenovic, G. Tsaramirsis, and P. G. Papageorgas, "Adaptive robust controller design-based RBF neural network for aerial robot arm model," *Electronics*, vol. 10, no. 7, p. 831, 2021.

[7] L. Zhou, S. Xu, H. Jin, and H. Jian, "A hybrid robust adaptive control for a quadrotor uav via mass observer and robust controller," *Advances in Mechanical Engineering*, vol. 13, no. 3, p. 16878140211002723, 2021.

[8] V. I. Utkin, *Sliding modes in control and optimization*. Springer Science & Business Media, 2013.

[9] J. Wang, P. Zhu, B. He, G. Deng, C. Zhang, and X. Huang, "An adaptive neural sliding mode control with ESO for uncertain nonlinear systems," *International Journal of Control, Automation and Systems*, vol. 19, no. 2, pp. 687–697, 2021.

[10] J. Li, J. Wang, H. Peng, Y. Hu, and H. Su, "Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[11] J. Y. Hung, W. Gao, and J. C. Hung, "Variable structure control: A survey," *IEEE Transactions on Industrial Electronics*, vol. 40, no. 1, pp. 2–22, 1993.

[12] W. Fang, F. Chao, C.-M. Lin, D. Zhou, L. Yang, X. Chang, Q. Shen, and C. Shang, "Visual-guided robotic object grasping using dual neural network controllers," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2282–2291, 2021.

[13] F. Chao, D. Zhou, C.-M. Lin, L. Yang, C. Zhou, and C. Shang, "Type-2 fuzzy hybrid controller network for robotic systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3778–3792, 2020.

[14] C.-M. Lin and C.-C. Chung, "Fuzzy brain emotional learning control system design for nonlinear systems," *International Journal of Fuzzy Systems*, vol. 17, no. 2, pp. 117–128, 2015.

[15] W. Fang, F. Chao, C.-M. Lin, L. Yang, C. Shang, and C. Zhou, "An improved fuzzy brain emotional learning model network controller for humanoid robots," *Frontiers in Neurorobotics*, vol. 13, p. 2, 2019.

[16] J. Zhang, Q. Li, X. Chang, F. Chao, C.-M. Lin, L. Yang, T. T. Huynh, L. Zheng, C. Zhou, and C. Shang, "A novel self-organizing emotional cmac network for robotic control," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–6.

[17] 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, 2019.

[18] J. Zhang, F. Chao, H. Zeng, C. Lin, and L. Yang, "A recurrent wavelet-based brain emotional learning network controller for nonlinear systems," *Soft Computing*, vol. 26, no. 6, pp. 3013–3028, 2022.

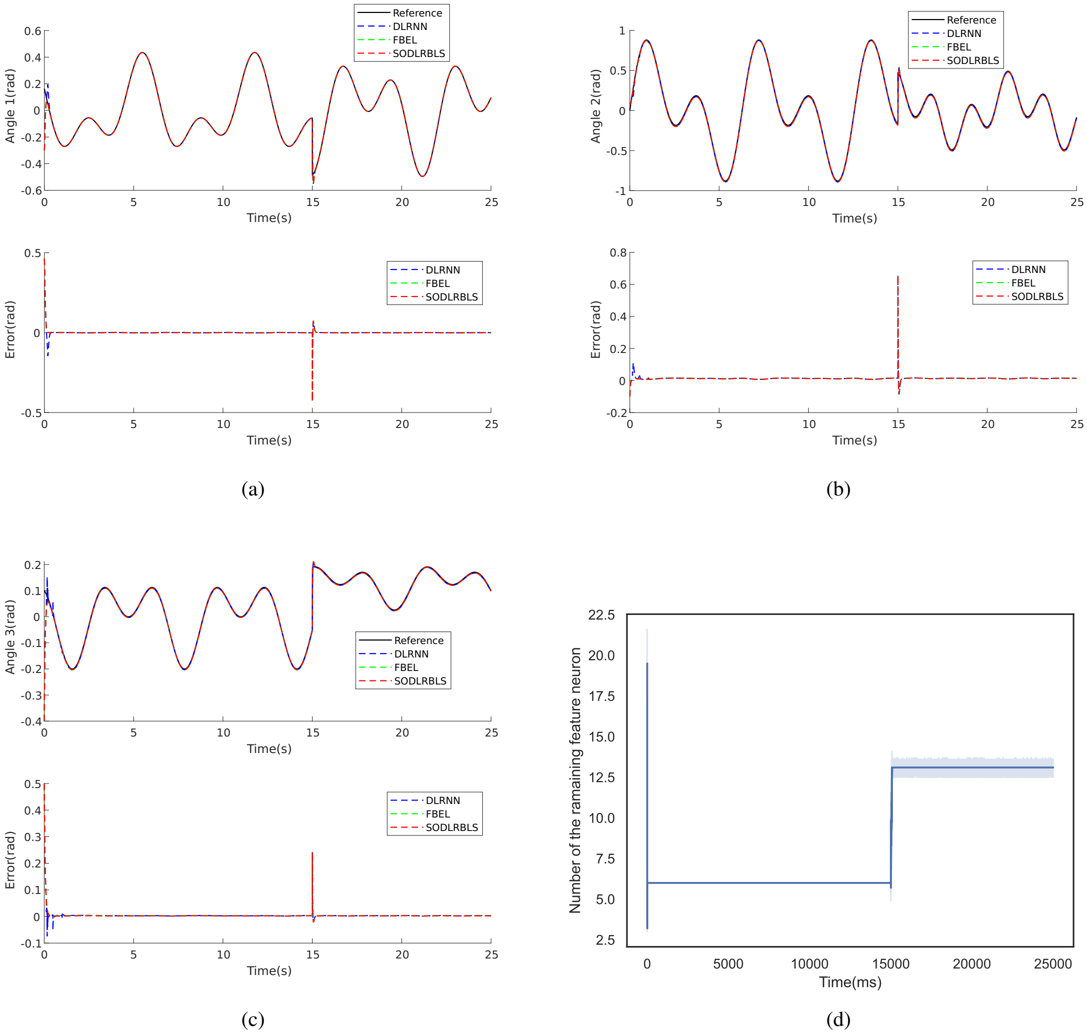


Fig. 5. The simulation results for a three-link robot manipulator. (a) Trajectory response and tracking error of joint 1, (b) Trajectory response and tracking error of joint 2, (c) Trajectory response and tracking error of joint 3, (d) The number of remaining feature nodes of SODLRBLS.

[19] C. Hwung and K. Yoon, "Multiple attribute decision-making: Methods and applications: A state-of-the-art-survey," 1981.

[20] H. Li, J. Huang, Y. Hu, S. Wang, J. Liu, and L. Yang, "A new tmy generation method based on the entropy-based topsis theory for different climatic zones in china," *Energy*, vol. 231, p. 120723, 2021.

[21] C.-N. Wang, T.-T. Dang, H. Tibo, and D.-H. Duong, "Assessing renewable energy production capabilities using dea window and fuzzy TOPSIS model," *Symmetry*, vol. 13, no. 2, p. 334, 2021.

[22] F. Foroozesh, S. M. Monavari, A. Salmanmahiny, M. Robati, and R. Rahimi, "Assessment of sustainable urban development based on a hybrid decision-making approach: Group fuzzy bwm, ahp, and topsis–gis," *Sustainable Cities and Society*, vol. 76, p. 103402, 2022.

[23] H.-S. Shih, H.-J. Shyur, and E. S. Lee, "An extension of topsis for group decision making," *Mathematical and Computer Modelling*, vol. 45, no. 7-8, pp. 801–813, 2007.

[24] C. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 10–24, 2017.

[25] C.-F. Hsu, B.-R. Chen, and B.-F. Wu, "Broad-learning recurrent hermite neural control for unknown nonlinear systems," *Knowledge-Based Systems*, vol. 242, p. 108263, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122000831

[26] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, 1992.

[27] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[28] J. Fei and C. Lu, "Adaptive sliding mode control of dynamic systems using double loop recurrent neural network structure," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1275–1286, 2017.

[29] C.-M. Lin and T.-T. Huynh, "Function-link fuzzy cerebellar model articulation controller design for nonlinear chaotic systems using TOPSIS multiple attribute decision-making method," *International Journal of Fuzzy Systems*, vol. 20, no. 6, pp. 1839–1856, 2018.

[30] R. V. Rao, *Decision making in the manufacturing environment: using graph theory and fuzzy multiple attribute decision making methods*. Springer, 2007, vol. 2.

[31] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[32] M. Zeleny, *Multiple criteria decision making Kyoto 1975*. Springer Science & Business Media, 2012, vol. 123.

[33] C.-H. Chen, C.-C. Chung, F. Chao, C.-M. Lin, and I. J. Rudas, "Intelligent robust control for uncertain nonlinear multivariable systems using recurrent cerebellar model neural networks," *Acta Polytechnica Hungarica*, vol. 12, no. 5, pp. 7–33, 2015.