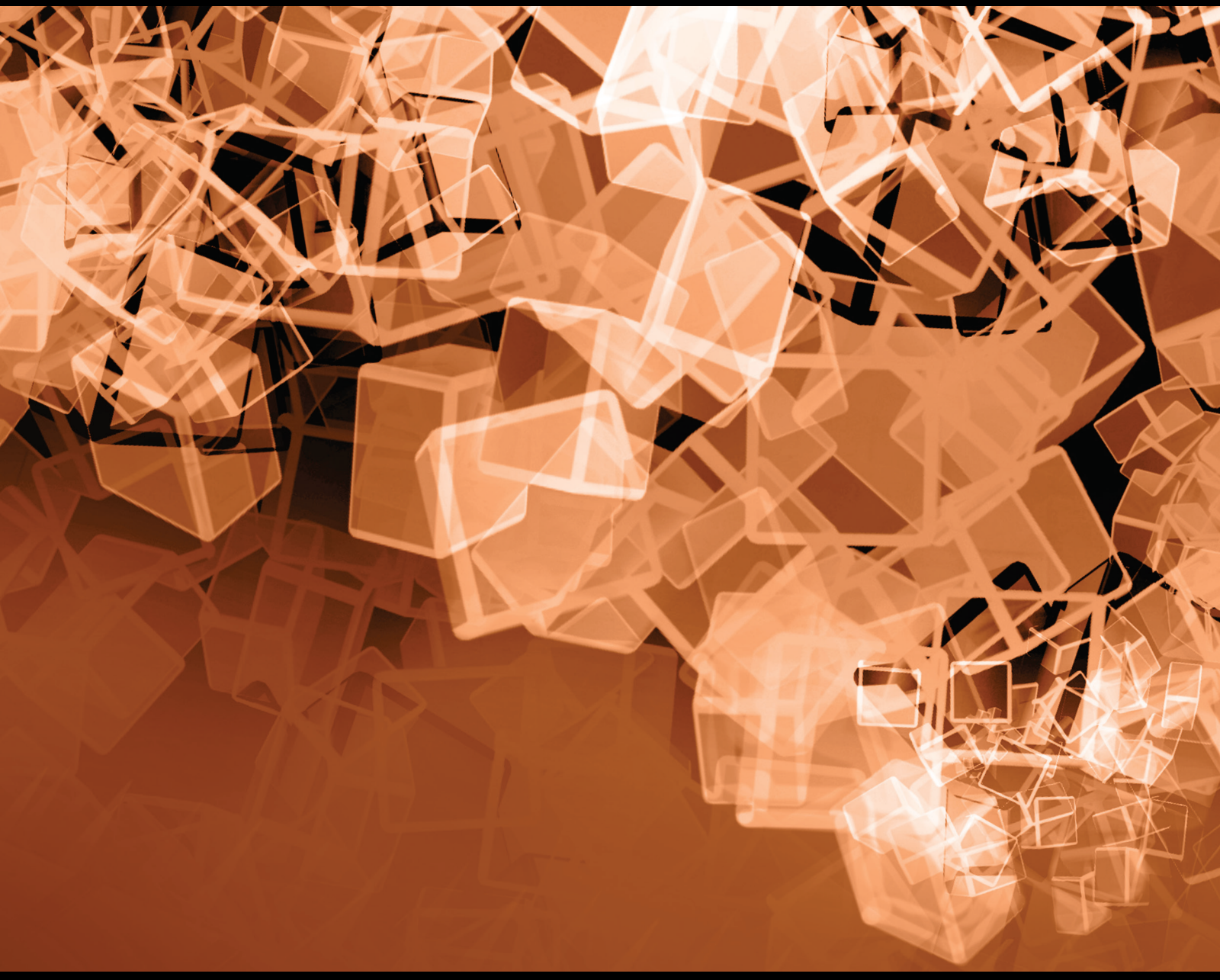
Discrete Dynamics in Nature and Society

Analysis, Control and Applications of Passivity in Complex Networks 2021

Lead Guest Editor: Jin-Liang Wang
Guest Editors: Guoguang Wen and Zi-Peng Wang



Analysis, Control and Applications of Passivity in Complex Networks 2021

Discrete Dynamics in Nature and Society

Analysis, Control and Applications of Passivity in Complex Networks 2021

Lead Guest Editor: Jin-Liang Wang
Guest Editors: Guoguang Wen and Zi-Peng Wang

Contents

Simulation and Computational Study of CFD on Tube MBR Membrane Assembly
Kun Feng, Chunqing Li, Ming Zhang, and Xuting Liu
Research Article (8 pages), Article ID 5577715, Volume 2021 (2021)

Adaptive Fuzzy Path Tracking Control for Mobile Robots with Unknown Control Direction
Qifei Du, Lin Sha, Wuxi Shi, and Liankun Sun
Research Article (7 pages), Article ID 9935271, Volume 2021 (2021)

Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 6920495, 9 pages
https://doi.org/10.1155/2022/6920495

Research Article

Feedback Control for Passivity of Memristor-Based Multiple Weighted Coupled Neural Networks

Xiang-Bo Wang,[1] **Hong-An Tang**,[1] **Qingling Xia**,[1] **Quanjun Zhao**,[2] **and Gang-Yi Tan**[1]

[1]*School of Artificial Intelligence, Chongqing University of Technology, Chongqing 401135, China*
[2]*School of Intelligent Manufacturing, Sichuan University of Arts and Science, Dazhou 635000, China*

Correspondence should be addressed to Hong-An Tang; tanghongan163@163.com and Qingling Xia; qingling@cqut.edu.cn

Received 19 May 2021; Accepted 24 November 2021; Published 1 February 2022

Academic Editor: A. E. Matouk

This paper investigates the passivity of multiple weighted coupled memristive neural networks (MWCMNNs) based on the feedback control. Firstly, a kind of memristor-based coupled neural network model with multiple weights is presented for the first time. Furthermore, a novel passivity criterion for MWCMNNs is established by constructing an appropriate Lyapunov functional and developing a suitable feedback controller. In addition, with the assistance of some inequality techniques, sufficient conditions for ensuring the input strict passivity and output strict passivity of MWCMNNs are derived. Finally, the validity of the theoretical results is verified by a numerical example.

1. Introduction

Neural networks (NNs) have aroused widespread attention since they have been applied in numerous fields including machine learning, deep learning, and engineering data prediction [1–3]. As the fourth two-terminal circuit element, the memristor was predicted to exist by Chua in 1971, and the prototype of memristor was obtained by the research team of HP for the first time [4–6]. Memristor is considered to be an excellent candidate for imitating biological synapses in circuit implementation of NNs owing to its characteristics of nanometer size, high storage capacity, and low energy consumption [7]. Through replacing the resistors with memristors in NNs circuit implementation, a new type of NN called memristive NN (MNN) has been successfully introduced [8]. Recently, it is reported that MNNs have many potential applications in face detection, bioengineering, pattern recognition, feature extraction, and associative memory [9–11]. To our knowledge, these applications for MNNs were to a great extent from the dynamical behaviors of MNNs. Particularly, the stability as a significant dynamical behavior for MNNs is one of the hot research topics [12]. Zhang et al. [12] obtained several sufficient conditions to insure the stability for MNNs.

The theory of passivity is valid and robust when studying the stability for nonlinear systems since passivity properties of a system can keep the internal stability of the system [13–15]. Up to now, a lot of interesting results about passivity of MNNs have been reported [16–18]. In [16], Meng and Xiang conducted the passivity analysis for a kind of complex-valued MNNs. Xiao et al. [17] obtained a new passivity criterion by utilizing set-valued mapping as well as transforming MNNs into traditional NNs. Based on the Lyapunov–Krasovskii method, Wu and Zeng [18] acquired an exponential passivity criterion for MNNs with mixed time-varying delays.

Complex networks (CNs) have attracted more and more interests of researchers in recent years, and CNs are ubiquitous in our life, such as communication networks, metabolic system networks, and food networks. Coupled NNs (CNNs) are a special class of CNs, which are composed of many NNs through mutually coupling [19, 20]. Considering the fact that the passivity of CNNs has been broadly applied to many fields including chaos generators and brain science.

Consequently, it is interesting to research the passivity for coupled MNNs (CMNNs) [21]. In [21], Yue et al. investigated the passivity of delayed CMNNs with reaction-diffusion terms with the aid of two pinning control schemes and some inequality techniques.

It is well known that most of networks in the practical world are supposed to be described as multiple weighted CNs, such as human social networks and public transport networks. However, only a few researchers have discussed multiple weighted CNNs (MWCNNs) in recent years [22]. Chen et al. [22] dealt with the dissipativity problem of MWCNNs via dynamic event-triggered pinning control. It should be noted that the passivity of multiple weighted CMNNs has only been discussed by few researchers so far.

It is worth noticing that the passivity of MNNs usually cannot be achieved on their own [23]. In consequence, it is essential to make use of some control strategies to make MNNs passive [24, 25]. To ensure exponential synchronization for MNNs, Lin et al. [24] developed a nonlinear feedback controller. Zhang et al. [25] derived some sufficient conditions for achieving finite time synchronization based on the feedback control. To our knowledge, the problem of passivity for multiple weighted CMNNs (MWCMNNs) under the feedback control has never been considered.

Motivated by the above analyses, this paper considers the passivity of MWCMNNs via the feedback control. The primary contributions of this paper are displayed as follows:

(1) A kind of memristor-based coupled neural network model with multiple weights is firstly proposed.

(2) It is first time that the feedback control strategy is adopted to ensure the passivity, output strict passivity, and input strict passivity of MWCMNNs.

(3) Several new passivity criteria are established according to linear matrix inequalities that can be checked through utilizing standard numerical packages.

2. Preliminaries

Let $\mathcal{N} = \{1, 2, \ldots, N\}, \mathbb{N} = \{1, 2, \ldots\}$, $\mathbb{R}^{m\times n}$ be the set of real matrices of order $m \times n$. $\mathbb{R}^{n\times n} \ni \Gamma > 0$ ($\mathbb{R}^{n\times n} \ni \Gamma < 0$) stands for that the matrix Γ is symmetric and positive (negative) definite. $\mathbb{R}^{n\times n} \ni \Gamma \geqslant 0$ ($\mathbb{R}^{n\times n} \ni \Gamma \leqslant 0$) stands for that the matrix Γ is symmetric and semipositive (seminegative) definite. Γ^T represents the transpose of matrix Γ. For any $\chi(t) = (\chi_1(t), \chi_2(t), \ldots, \chi_n(t)) \in \mathbb{R}^n$, $\|\chi(t)\|_2 = \sqrt{\chi^T(t)\chi(t)}$. $\lambda_m(\Gamma)$ and $\lambda_M(\Gamma)$ mean the minimal as well as the maximal eigenvalue of matrix Γ, respectively.

Definition 1 (see [26]). A system with supply rate $\mathscr{W}(u, y)$ is dissipative if there is a nonnegative storage function $\mathscr{V}$: $[0, +\infty) \longrightarrow [0, +\infty)$, such that

$$\int_{t_0}^{t_\varepsilon} \mathscr{W}(u(t), y(t))dt \geq \mathscr{V}(t_\varepsilon) - \mathscr{V}(t_0), \tag{1}$$

for any $t_0, t_\varepsilon \in [0, +\infty)$ and $t_\varepsilon \geq t_0$, where $\mathbb{R}^w \ni y(t)$, $\mathbb{R}^m \ni u(t)$ are the output and input of the system, respectively.

Definition 2 (see [27]). If a system is dissipative and satisfying

$$\mathscr{W}(u(t), y(t)) = y^T(t)Zu(t), \tag{2}$$

in which $Z \in \mathbb{R}^{w\times m}$ is a constant matrix, the system can achieve the passivity.

Definition 3 (see [27]). If a system is dissipative and satisfying

$$\mathscr{W}(u(t), y(t)) = y^T(t)Zu(t) - y^T(t)\Phi_1 y(t) - u^T(t)\Phi_2 u(t), \tag{3}$$

in which $Z \in \mathbb{R}^{w\times m}$, $0 \leq \Phi_1 \in \mathbb{R}^{w\times w}$, $0 \leq \Phi_2 \in \mathbb{R}^{m\times m}$, and $\lambda_m(\Phi_1) + \lambda_m(\Phi_2) > 0$, then the system can achieve the strict passivity.

If $0 \leq \Phi_1$, the system is output-strictly passive, and if $0 \leq \Phi_2$, the system is input-strictly passive.

3. Passivity of MWCMNNs

3.1. Network Model. The model of coupled neural network with multiple weights is given by

$$\begin{aligned}\dot{x}_i(t) = &-Kx_i(t) + Af(x_i(t)) + \sum_{r=1}^{s}\sum_{j=1}^{N} d_r G_{ij}^r H^r x_j(t) \\ &+ J + Cu_i(t),\end{aligned} \tag{4}$$

where $i = 1, 2, \ldots, N$, $x_i(t) = (x_{i1}(t), x_{i2}(t), \ldots, x_{in}(t))^T \in \mathbb{R}^n$ indicates the state vector of the ith node; $0 < K = \operatorname{diag}(k_1, k_2, \ldots, k_n) \in \mathbb{R}^{n\times n}$; $f(x_i(t)) = (f_1(x_{i1}(t)), f_2(x_{i2}(t)), \ldots, f_n(x_{in}(t)))^T \in \mathbb{R}^n$, and $f_\rho(\cdot)$ means the activation function of ρth neuron; $\mathbb{R}^{n\times n} \ni A$ denotes a constant matrix; $J = (J_1, J_2, \ldots, J_n)^T \in \mathbb{R}^n$ is the constant external input vector; $C \in \mathbb{R}^{n\times m}$ is a constant matrix; $u_i(t) = (u_{i1}(t), u_{i2}(t), \ldots, u_{im}(t))^T \in \mathbb{R}^m$ denotes the external input vector; $0 < d_r$ stands for the coupling strength of the rth coupling form; $\mathbb{R}^{n\times n} \ni H^r = \operatorname{diag}(h_1^r, h_2^r, \ldots, h_n^r) > 0$ means the inner coupling matrix in the rth coupling form; $G^r = (G_{ij}^r)_{N\times N}$ is the external coupling matrix for the rth coupling form, where G_{ij}^r satisfies the following conditions:

$$\begin{aligned} &G_{ij}^r = G_{ji}^r \geq 0, \quad i \neq j; \\ &G_{ii}^r = -\sum_{\substack{j=1 \\ j\neq i}}^{N} G_{ij}^r, \end{aligned} \tag{5}$$

if there is a link between node i and node j, then $G_{ij}^r > 0$ or else $G_{ij}^r = 0$.

Consider the following multiple weighted coupled memristive neural network (MWCMNN) consisting of N identical MNNs with multiple weights:

$$\begin{aligned}\dot{x}_i(t) = &-Kx_i(t) + A(x_i(t))f(x_i(t)) \\ &+ B(x_i(t))g(x_i(t-\tau)) + Cu_i(t) \\ &+ \sum_{r=1}^{s}\sum_{j=1}^{N} d_r G_{ij}^r H^r x_j(t) \\ &+ J + \psi_i(t), \quad i = 1, 2, \ldots, N,\end{aligned} \tag{6}$$

where $K, J, x_i(t), d_r, G_{ij}^r, H^r, C, u_i(t), f(x_i(t))$ have the same meanings as in network (1); $A(x_i(t)) = (a_{\eta\rho}(x_{i\eta}(t)))_{n\times n}$, $B(x_i(t)) = (b_{\eta\rho}(x_{i\eta}(t)))_{n\times n}$, $\eta = 1, 2, \ldots, n, \rho = 1, 2, \ldots, n$; $x_{i\eta}(t) \in \mathbb{R}$ stands for the voltage for capacitor $\mathscr{C}_\eta$; $x_i(t-\tau) = (x_{i1}(t-\tau), x_{i2}(t-\tau), \ldots, x_{in}(t-\tau))^T \in \mathbb{R}^n$; $g(x_i(t-\tau)) = (g_1(x_{i1}(t-\tau)), g_2(x_{i2}(t-\tau)), \ldots, g_n(x_{in}(t-\tau)))^T \in \mathbb{R}^n$, and $g_\rho(\cdot)$ means the activation function of ρth neuron; τ indicates the propagation delay; $\psi_i(t) \in \mathbb{R}^n$ is the control input; and $a_{\eta\rho}(x_{i\eta}(t))b_{\eta\rho}(x_{i\eta}(t))$ are described by

$$\begin{aligned} a_{\eta\rho}(x_{i\eta}(t)) &= \frac{\mathscr{W}_{\eta\rho}}{\mathscr{C}_\eta} \times \mathrm{sign}_{\eta\rho}, \\ b_{\eta\rho}(x_{i\eta}(t)) &= \frac{\mathscr{M}_{\eta\rho}}{\mathscr{C}_\eta} \times \mathrm{sign}_{\eta\rho}, \\ \mathrm{sign}_{\eta\rho} &= \begin{cases} 1, & \eta \neq \rho, \\ -1, & \eta = \rho, \end{cases}\end{aligned} \tag{7}$$

where $\mathscr{W}_{\eta\rho}$ and $\mathscr{M}_{\eta\rho}$ represent the memductances of memristors $\mathscr{A}_{\eta\rho}$ and $\mathscr{B}_{\eta\rho}$, respectively. $\mathscr{A}_{\eta\rho}$ indicates the memristor between $x_{i\eta}(t)$ and the function $f_\rho(x_{i\rho}(t))$, and $\mathscr{B}_{\eta\rho}$ indicates the memristor between $x_{i\eta}(t)$ and the function $g_\rho(x_{i\rho}(t-\tau))$. In the light of the traits of voltage and current of memristor, we can obtain that

$$\begin{aligned} a_{\eta\rho}(x_{i\eta}(t)) &= \begin{cases} \hat{a}_{\eta\rho}, & |x_{i\eta}(t)| \le \Psi_\eta, \\ \check{a}_{\eta\rho}, & |x_{i\eta}(t)| > \Psi_\eta, \end{cases} \\ b_{\eta\rho}(x_{i\eta}(t)) &= \begin{cases} \hat{b}_{\eta\rho}, & |x_{i\eta}(t)| \le \Psi_\eta, \\ \check{b}_{\eta\rho}, & |x_{i\eta}(t)| > \Psi_\eta, \end{cases}\end{aligned} \tag{8}$$

where the switching jumps $\Psi_\eta > 0$; $\check{a}_{\eta\rho}, \hat{a}_{\eta\rho}, \check{b}_{\eta\rho}, \hat{b}_{\eta\rho}$ are constants, $\eta, \rho = 1, 2, \ldots, n$.

Define $\overline{a}_{\eta\rho} = |\check{a}_{\eta\rho} - \hat{a}_{\eta\rho}|$, $\overline{A} = (\overline{a}_{\eta\rho})_{n\times n}$, $\widetilde{a}_{\eta\rho} = \max\{|\check{a}_{\eta\rho}|, |\hat{a}_{\eta\rho}|\}$, $\widetilde{A} = \mathrm{diag}\,(\sum_{\rho=1}^n \widetilde{a}_{1\rho}^2, \sum_{\rho=1}^n \widetilde{a}_{2\rho}^2, \ldots, \sum_{\rho=1}^n \widetilde{a}_{n\rho}^2)$, $\overline{b}_{\eta\rho} = |\check{b}_{\eta\rho} - \hat{b}_{\eta\rho}|$, $\overline{B} = (\overline{b}_{\eta\rho})_{n\times n}$, $\widetilde{b}_{\eta\rho} = \max\{|\check{b}_{\eta\rho}|, |\hat{b}_{\eta\rho}|\}$, $\widetilde{B} = \mathrm{diag}\,(\sum_{\rho=1}^n \widetilde{b}_{1\rho}^2, \sum_{\rho=1}^n \widetilde{b}_{2\rho}^2, \ldots, \sum_{\rho=1}^n \widetilde{b}_{n\rho}^2)$.

Remark 1. Considering the fact that plenty of networks in the actual world ought to be described by CNs with multiple weights, for instance, human social networks and public transport networks. Nevertheless, only some researchers have investigated the passivity of MWCNNs [22]. Therefore, it is interesting to discuss the passivity of MWCMNNs. In this paper, a kind of memristor-based coupled neural network model with multiple weights is proposed.

Throughout this paper, we put forward the following assumption.

Assumption 1. If there are positive constants $\lambda_\rho, \mu_\rho, \widetilde{\lambda}_\rho$, and $\widetilde{\mu}_\rho, \rho = 1, 2, \ldots, n$, such that

$$\begin{aligned} |f_\rho(\delta_1) - f_\rho(\delta_2)| &\le \lambda_\rho|\delta_1 - \delta_2|, \quad |f_\rho(\delta)| \le \widetilde{\lambda}_\rho, \\ |g_\rho(\delta_1) - g_\rho(\delta_2)| &\le \mu_\rho|\delta_1 - \delta_2|, \quad |g_\rho(\delta)| \le \widetilde{\mu}_\rho,\end{aligned} \tag{9}$$

for any $\delta, \delta_1, \delta_2 \in \mathbb{R}$.

Consider that $x^* = (x_1^*, x_2^*, \ldots, x_n^*)^T \in \mathbb{R}^n$ is an equilibrium point for network (6), then

$$-Kx^* + A(x^*)f(x^*) + B(x^*)g(x^*) + J = 0. \tag{10}$$

Letting error vector $z_i(t) = x_i(t) - x^*$, we get

$$\begin{aligned}\dot{z}_i(t) = &-Kz_i(t) + A(x_i(t))f(z_i(t)) \\ &+ [A(x_i(t)) - A(x^*)]f(x^*) \\ &+ B(x_i(t))g(z_i(t-\tau)) \\ &+ [B(x_i(t)) - B(x^*)]g(x^*) + Cu_i(t) \\ &+ \sum_{r=1}^{s}\sum_{j=1}^{N} d_r G_{ij}^r H^r z_j(t) + \psi_i(t),\end{aligned} \tag{11}$$

where

$$\begin{aligned} &i = 1, 2, \ldots, N, z_i(t-\tau) = (z_{i1}(t-\tau), z_{i2}(t-\tau), \ldots, z_{in}(t-\tau))^T, \\ &f(z_i(t)) = f(x_i(t)) - f(x^*),\ g(z_i(t-\tau)) = g(x_i(t-\tau)) - g(x^*).\end{aligned} \tag{12}$$

According to the network (11), a feedback controller is developed as follows:

$$\psi_i(t) = -\mathrm{sign}(z_i(t))(\overline{A}\widetilde{\lambda} + \overline{B}\widetilde{\mu}), \tag{13}$$

where $\widetilde{\lambda} = (\widetilde{\lambda}_1, \widetilde{\lambda}_2, \ldots, \widetilde{\lambda}_n)^T$, $\widetilde{\mu} = (\widetilde{\mu}_1, \widetilde{\mu}_2, \ldots, \widetilde{\mu}_n)^T$, $\mathrm{sign}(z_i(t)) = \mathrm{diag}(\mathrm{sign}(z_{i1}(t)), \mathrm{sign}(z_{i2}(t)), \ldots, \mathrm{sign}(z_{in}(t)))$, $i = 1, 2, \ldots, N$.

From (11) and (13), one has

$$\begin{aligned}\dot{z}_i(t) = &-Kz_i(t) + A(x_i(t))f(z_i(t)) \\ &+ [A(x_i(t)) - A(x^*)]f(x^*) \\ &+ B(x_i(t))g(z_i(t-\tau)) \\ &+ [B(x_i(t)) - B(x^*)]g(x^*) + Cu_i(t) \\ &+ \sum_{r=1}^{s}\sum_{j=1}^{N} d_r G_{ij}^r H^r z_j(t) - \operatorname{sign}(z_i(t))(\overline{A}\widetilde{\lambda} + \overline{B}\widetilde{\mu}).\end{aligned} \tag{14}$$

The output vector $y_i(t) \in \mathbb{R}^w$ of the network (14) is given by

$$y_i(t) = Q_1 z_i(t) + Q_2 u_i(t), \tag{15}$$

in which $Q_1 \in \mathbb{R}^{w\times n}$ and $Q_2 \in \mathbb{R}^{w\times m}$.

For convenience, we denote

$$\begin{aligned}\wedge &= \operatorname{diag}\left(\lambda_1^2, \lambda_2^2, \ldots, \lambda_n^2\right), \\ M &= \operatorname{diag}\left(\mu_1^2, \mu_2^2, \ldots, \mu_n^2\right), \\ z(t) &= \left(z_1^T(t), z_2^T(t), \ldots, z_N^T(t)\right)^T, \\ u(t) &= \left(u_1^T(t), u_2^T(t), \ldots, u_N^T(t)\right)^T, \\ y(t) &= \left(y_1^T(t), y_2^T(t), \ldots, y_N^T(t)\right)^T.\end{aligned} \tag{16}$$

3.2. Passivity Criteria

Theorem 1. *If there is a matrix $F \in \mathbb{R}^{wN\times mN}$ satisfying*

$$\begin{pmatrix} \Xi_1 & \Xi_2 \\ \Xi_2^T & -\frac{1}{2}\left[\left(I_N \otimes Q_2^T\right)F + F^T\left(I_N \otimes Q_2\right)\right] \end{pmatrix} \le 0, \tag{17}$$

where $\Xi_1 = I_N \otimes (-2K + \wedge + \widetilde{A} + M + \widetilde{B}) + 2\sum_{r=1}^{s} d_r G^r \otimes H^r$, $\Xi_2 = I_N \otimes C - 1/2\,(I_N \otimes Q_1^T)F$, *then the network (14) is passive.*

Proof. Select the following Lyapunov functional for the network (14):

$$V(t) = \sum_{i=1}^{N} z_i^T(t) z_i(t) + \sum_{i=1}^{N}\int_{t-\tau}^{t} z_i^T(s) M z_i(s)\,\mathrm{d}s. \tag{18}$$

Then, one has

$$\begin{aligned}D^+V(t) = &\sum_{i=1}^{N} z_i^T(t) M z_i(t) - \sum_{i=1}^{N} z_i^T(t-\tau) M z_i(t-\tau) \\ &+ 2\sum_{i=1}^{N} z_i^T(t)\dot{z}_i(t) \\ \le\, & 2\sum_{i=1}^{N} z_i^T(t)\{-Kz_i(t) + A(x_i(t))f(z_i(t)) \\ &+ [A(x_i(t)) - A(x^*)]f(x^*) \\ &+ B(x_i(t))g(z_i(t-\tau)) \\ &+ [B(x_i(t)) - B(x^*)]g(x^*) + Cu_i(t) \\ &+ \sum_{r=1}^{s}\sum_{j=1}^{N} d_r G_{ij}^r H^r z_j(t) \\ &- \operatorname{sign}(z_i(t))(\overline{A}\widetilde{\lambda} + \overline{B}\widetilde{\mu})\} \\ &+ \sum_{i=1}^{N} z_i^T(t) M z_i(t) \\ &- \sum_{i=1}^{N} z_i^T(t-\tau) M z_i(t-\tau).\end{aligned} \tag{19}$$

Based on Assumption 1, one can obtain

$$\begin{aligned}&2z_i^T(t)[A(x_i(t)) - A(x^*)]f(x^*) \\ &= 2\sum_{\eta=1}^{n}\sum_{\rho=1}^{n} z_{i\eta}(t)\left(a_{\eta\rho}\left(x_{i\eta}(t)\right) - a_{\eta\rho}\left(x_\eta^*\right)\right)f_\rho\left(x_\rho^*\right) \\ &\le 2\sum_{\eta=1}^{n}\sum_{\rho=1}^{n}\left|z_{i\eta}(t)\right|\left|\check{a}_{\eta\rho} - \widehat{a}_{\eta\rho}\right|\widetilde{\lambda}_\rho \\ &= 2\left|z_i^T(t)\right|\overline{A}\widetilde{\lambda}.\end{aligned} \tag{20}$$

Similarly, one has

$$\begin{aligned}&2z_i^T(t)[B(x_i(t)) - B(x^*)]g(x^*) \\ &= 2\sum_{\eta=1}^{n}\sum_{\rho=1}^{n} z_{i\eta}(t)\left(b_{\eta\rho}\left(x_{i\eta}(t)\right) - b_{\eta\rho}\left(x_\eta^*\right)\right)g_\rho\left(x_\rho^*\right) \\ &\le 2\sum_{\eta=1}^{n}\sum_{\rho=1}^{n}\left|z_{i\eta}(t)\right|\left|\check{b}_{\eta\rho} - \widehat{b}_{\eta\rho}\right|\widetilde{\mu}_\rho \\ &= 2\left|z_i^T(t)\right|\overline{B}\widetilde{\mu}.\end{aligned} \tag{21}$$

It can be obtained from Lemma 2.1in [28] and Assumption 1 that

$$\begin{aligned}
&2z_i^T(t)A(x_i(t))f(z_i(t))\\
&\le z_i^T(t)A(x_i(t))A^T(x_i(t))z_i(t)+f^T(z_i(t))f(z_i(t))\\
&=\sum_{\eta=1}^{n}\sum_{\rho=1}^{n}z_{i\eta}^2(t)a_{\eta\rho}^2\left(x_{i\eta}(t)\right)+\sum_{\rho=1}^{n}f_\rho^2\left(z_{i\rho}(t)\right)\\
&\le\sum_{\eta=1}^{n}\sum_{\rho=1}^{n}z_{i\eta}^2(t)\tilde{a}_{\eta\rho}^2+\sum_{\rho=1}^{n}\lambda_\rho^2z_{i\rho}^2(t)\\
&=z_i^T(t)\tilde{A}z_i(t)+z_i^T(t)\wedge z_i(t).
\end{aligned}\tag{22}$$

Similarly, we get

$$\begin{aligned}
&2z_i^T(t)B(x_i(t))g(z_i(t-\tau))\\
&\le z_i^T(t)B(x_i(t))B^T(x_i(t))z_i(t)\\
&\quad+g^T(z_i(t-\tau))g(z_i(t-\tau))\\
&=\sum_{\eta=1}^{n}\sum_{\rho=1}^{n}z_{i\eta}^2(t)b_{\eta\rho}^2\left(x_{i\eta}(t)\right)+\sum_{\rho=1}^{n}g_\rho^2\left(z_{i\rho}(t-\tau)\right)\\
&\le\sum_{\eta=1}^{n}\sum_{\rho=1}^{n}z_{i\eta}^2(t)\tilde{b}_{\eta\rho}^2+\sum_{\rho=1}^{n}\mu_\rho^2z_{i\rho}^2(t-\tau)\\
&=z_i^T(t)\tilde{B}z_i(t)+z_i^T(t-\tau)Mz_i(t-\tau).
\end{aligned}\tag{23}$$

According to (19)–(23), one gets

$$\begin{aligned}
D^+V(t)&\le\sum_{i=1}^{N}z_i^T(t)(-2K+\wedge+\tilde{A}+M+\tilde{B})z_i(t)\\
&\quad+2\sum_{i=1}^{N}z_i^T(t)Cu_i(t)\\
&\quad+2\sum_{r=1}^{s}\sum_{i=1}^{N}\sum_{j=1}^{N}d_rG_{ij}^rz_i^T(t)H^rz_j(t)\\
&=\sum_{i=1}^{N}z_i^T(t)(-2K+\wedge+\tilde{A}+M+\tilde{B})z_i(t)\\
&\quad+2\sum_{i=1}^{N}z_i^T(t)Cu_i(t)\\
&\quad+2\sum_{r=1}^{s}d_rz^T(t)(G^r\otimes H^r)z(t)\\
&=z^T(t)\Big[I_N\otimes(-2K+\wedge+\tilde{A}+M+\tilde{B})\\
&\quad+2\sum_{r=1}^{s}d_rG^r\otimes H^r\Big]z(t)\\
&\quad+2z^T(t)(I_N\otimes C)u(t).
\end{aligned}\tag{24}$$

Further, one obtains

$$\begin{aligned}
&D^+V(t)-y^T(t)Fu(t)\\
&\le z^T(t)\Big[I_N\otimes(-2K+\wedge+\tilde{A}+M+\tilde{B})\\
&\quad+2\sum_{r=1}^{s}d_rG^r\otimes H^r\Big]z(t)\\
&\quad+z^T(t)\left[2I_N\otimes C-\left(I_N\otimes Q_1^T\right)F\right]u(t)\\
&\quad-u^T(t)\left\{\frac{1}{2}\left[\left(I_N\otimes Q_2^T\right)F+F^T\left(I_N\otimes Q_2\right)\right]\right\}u(t)\\
&=\vartheta^T(t)\begin{pmatrix}\Xi_1 & \Xi_2\\ \Xi_2^T & -\frac{1}{2}\left[\left(I_N\otimes Q_2^T\right)F+F^T\left(I_N\otimes Q_2\right)\right]\end{pmatrix}\vartheta(t),
\end{aligned}\tag{25}$$

where $\vartheta(t)=(z^T(t),u^T(t))^T$.

From (17), one can derive

$$y^T(t)Fu(t)\ge D^+V(t).\tag{26}$$

By (26), one has

$$\int_{t_0}^{t_\varepsilon}y^T(s)Fu(s)ds\ \ge V(t_\varepsilon)-V(t_0),\tag{27}$$

for any $t_0,t_\varepsilon\in[0,+\infty)$ and $t_\varepsilon\ge t_0$. □

Theorem 2. *If there exist matrices $F\in\mathbb{R}^{wN\times mN}$ and $0<L_1\in\mathbb{R}^{mN\times mN}$ satisfying*

$$\begin{pmatrix}\Xi_1 & \Xi_2\\ \Xi_2^T & -\frac{1}{2}\left[\left(I_N\otimes Q_2^T\right)F+F^T\left(I_N\otimes Q_2\right)\right]+L_1\end{pmatrix}\le 0,\tag{28}$$

where $\Xi_1=I_N\otimes(-2K+\wedge+\tilde{A}+M+\tilde{B})\ +2\sum_{r=1}^{s}d_rG^r\otimes H^r$, $\Xi_2=I_N\otimes C-1/2(I_N\otimes Q_1^T)F$, then the network (14) is input-strictly passive.

Proof. For the network (14), choosing the identical Lyapunov functional as (18), then one can obtain

$$\begin{aligned}
&D^+V(t)-y^T(t)Fu(t)+u^T(t)L_1u(t)\\
&\le z^T(t)\Big[I_N\otimes(-2K+\wedge+\tilde{A}+M+\tilde{B})\\
&\quad+2\sum_{r=1}^{s}d_rG^r\otimes H^r\Big]z(t)\\
&\quad+z^T(t)\left[2I_N\otimes C-\left(I_N\otimes Q_1^T\right)F\right]u(t)\\
&\quad-u^T(t)\left\{\frac{1}{2}\left[\left(I_N\otimes Q_2^T\right)F+F^T\left(I_N\otimes Q_2\right)\right]-L_1\right\}u(t)\\
&=\vartheta^T(t)\begin{pmatrix}\Xi_1 & \Xi_2\\ \Xi_2^T & -\frac{1}{2}\left[\left(I_N\otimes Q_2^T\right)F+F^T\left(I_N\otimes Q_2\right)\right]+L_1\end{pmatrix}\vartheta(t),
\end{aligned}\tag{29}$$

where $\vartheta(t)=(z^T(t),u^T(t))^T$.

From (28), we can get

$$y^T(t)Fu(t) - u^T(t)L_1u(t) \geq D^+V(t). \tag{30}$$

By (30), one has

$$\int_{t_0}^{t_\varepsilon}\left(y^T(s)Fu(s) - u^T(s)L_1u(s)\right)ds \geq V(t_\varepsilon) - V(t_0), \tag{31}$$

for any $t_0, t_\varepsilon \in [0, +\infty)$ and $t_\varepsilon \geq t_0$. □

Theorem 3. *If there exist matrices $F \in \mathbb{R}^{wN\times mN}$ and $0 < L_2 \in \mathbb{R}^{wN\times wN}$ satisfying*

$$\begin{pmatrix} \Xi_3 & \Xi_4 \\ \Xi_4^T & \Xi_5 \end{pmatrix} \leq 0, \tag{32}$$

where

$$\begin{aligned}
\Xi_3 &= I_N \otimes (-2K + \wedge + \widetilde{A} + M + \widetilde{B}) + 2\sum_{r=1}^{s} d_r G^r \otimes H^r + \left(I_N \otimes Q_1^T\right)L_2\left(I_N \otimes Q_1\right),\\
\Xi_4 &= I_N \otimes C - \frac{1}{2}\left(I_N \otimes Q_1^T\right)F + \left(I_N \otimes Q_1^T\right)L_2\left(I_N \otimes Q_2\right),\\
\Xi_5 &= -\frac{1}{2}\left[\left(I_N \otimes Q_2^T\right)F + F^T\left(I_N \otimes Q_2\right)\right] + \left(I_N \otimes Q_2^T\right)L_2\left(I_N \otimes Q_2\right),
\end{aligned} \tag{33}$$

then the network (14) is output-strictly passive.

Proof. For the network (14), choosing the identical Lyapunov functional as (18), then we can derive

$$\begin{aligned}
&D^+V(t) - y^T(t)Fu(t) + y^T(t)L_2y(t)\\
&\leq z^T(t)\Big[I_N \otimes (-2K + \wedge + \widetilde{A} + M + \widetilde{B})\\
&\quad + 2\sum_{r=1}^{s} d_r G^r \otimes H^r + \left(I_N \otimes Q_1^T\right)L_2\left(I_N \otimes Q_1\right)\Big]z(t)\\
&\quad + z^T(t)\Big[2I_N \otimes C - \left(I_N \otimes Q_1^T\right)F\\
&\quad + 2\left(I_N \otimes Q_1^T\right)L_2\left(I_N \otimes Q_2\right)\Big]u(t)\\
&\quad - u^T(t)\Big\{\frac{1}{2}\left[\left(I_N \otimes Q_2^T\right)F + F^T\left(I_N \otimes Q_2\right)\right]\\
&\quad - \left(I_N \otimes Q_2^T\right)L_2\left(I_N \otimes Q_2\right)\Big\}u(t)\\
&= \vartheta^T(t)\begin{pmatrix} \Xi_3 & \Xi_4 \\ \Xi_4^T & \Xi_5 \end{pmatrix}\vartheta(t),
\end{aligned} \tag{34}$$

where $\vartheta(t) = (z^T(t), u^T(t))^T$.

From (32), one can get

$$y^T(t)Fu(t) - y^T(t)L_2y(t) \geq D^+V(t). \tag{35}$$

By (35), one has

$$\int_{t_0}^{t_\varepsilon}\left(y^T(s)Fu(s) - y^T(s)L_2y(s)\right)ds \geq V(t_\varepsilon) - V(t_0), \tag{36}$$

for any $t_0, t_\varepsilon \in [0, +\infty)$ and $t_\varepsilon \geq t_0$. □

Remark 2. It is a key issue that NNs are unable to achieve the passivity by themselves in some circumstances [23]. As a consequence, it is necessary to utilize an appropriate control method to make NNs passive. For all we know, the passivity problem of MWCMNNs via feedback control has not been researched. In the above discussion, with the help of a developed feedback controller, several criteria are established to ensure that the proposed network is passive, output-strictly passive, and input-strictly passive, respectively.

4. Numerical Example

Example 1. The MWCMNN is considered as follows:

$$\begin{aligned}\dot{x}_i(t) = &-Kx_i(t) + A(x_i(t))f(x_i(t)) \\ &+ B(x_i(t))g(x_i(t-\tau)) + Cu_i(t) \\ &+ \sum_{r=1}^{3}\sum_{j=1}^{6} d_r G_{ij}^r H^r x_j(t) + J + \psi_i(t),\end{aligned} \tag{37}$$

in which $i = 1, 2, \ldots, 6$, $f_\rho(\delta) = g_\rho(\delta) = 1/8(|\delta+1| - |\delta-1|)$, $\rho = 1, 2, 3$, $K = \operatorname{diag}(1.6, 2.1, 2.6)$, $\tau = 1$, $J = (0,0,0)^T$, $d_1 = 0.1$, $d_2 = 0.2$, $d_3 = 0.3$, $H^1 = \operatorname{diag}(0.3, 0.6, 0.4)$, $H^2 = \operatorname{diag}(0.5, 0.6, 0.2)$, $H^3 = \operatorname{diag}(0.6, 0.5, 0.3)$, and the matrices $G^1, G^2, G^3, C, A(x_i(t)), B(x_i(t))$ are chosen as follows:

$$G^1 = \begin{pmatrix} -0.2 & 0 & 0.1 & 0 & 0.1 & 0 \\ 0 & -0.4 & 0 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0 & -0.5 & 0 & 0.2 & 0.2 \\ 0 & 0.2 & 0 & -0.5 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.1 & -0.7 & 0.2 \\ 0 & 0.1 & 0.2 & 0.2 & 0.2 & -0.7 \end{pmatrix},$$

$$G^2 = \begin{pmatrix} -0.3 & 0 & 0.1 & 0 & 0.1 & 0.1 \\ 0 & -0.5 & 0 & 0.2 & 0.1 & 0.2 \\ 0.1 & 0 & -0.4 & 0 & 0.3 & 0 \\ 0 & 0.2 & 0 & -0.6 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.3 & 0.2 & -0.9 & 0.2 \\ 0.1 & 0.2 & 0 & 0.2 & 0.2 & -0.7 \end{pmatrix},$$

$$G^3 = \begin{pmatrix} -0.4 & 0.1 & 0.1 & 0 & 0.1 & 0.1 \\ 0.1 & -0.4 & 0 & 0.1 & 0.2 & 0 \\ 0.1 & 0 & -0.7 & 0.2 & 0.3 & 0.1 \\ 0 & 0.1 & 0.2 & -0.5 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0.1 & -1.0 & 0.3 \\ 0.1 & 0 & 0.1 & 0.1 & 0.3 & -0.6 \end{pmatrix},$$

$$C = \begin{pmatrix} 0.3 & 0.6 \\ 0.3 & 0.1 \\ 0.2 & 0.4 \end{pmatrix}, \tag{38}$$

$$\begin{aligned}
a_{11}(x_{i1}(t)) &= \begin{cases} -0.20, |x_{i1}(t)| \le 1, \\ -0.36, |x_{i1}(t)| > 1, \end{cases} \\
a_{12}(x_{i1}(t)) &= \begin{cases} 0.43, |x_{i1}(t)| \le 1, \\ -0.20, |x_{i1}(t)| > 1, \end{cases} \\
a_{13}(x_{i1}(t)) &= \begin{cases} -0.20, |x_{i1}(t)| \le 1, \\ 0.32, |x_{i1}(t)| > 1, \end{cases} \\
a_{21}(x_{i2}(t)) &= \begin{cases} 0.47, |x_{i2}(t)| \le 1, \\ -0.30, |x_{i2}(t)| > 1, \end{cases} \\
a_{22}(x_{i2}(t)) &= \begin{cases} -0.53, |x_{i2}(t)| \le 1, \\ 0.40, |x_{i2}(t)| > 1, \end{cases} \\
a_{23}(x_{i2}(t)) &= \begin{cases} 0.33, |x_{i2}(t)| \le 1, \\ -0.66, |x_{i2}(t)| > 1, \end{cases} \\
a_{31}(x_{i3}(t)) &= \begin{cases} 0.56, |x_{i3}(t)| \le 1, \\ 0.40, |x_{i3}(t)| > 1, \end{cases} \\
a_{32}(x_{i3}(t)) &= \begin{cases} 0.20, |x_{i3}(t)| \le 1, \\ -0.42, |x_{i3}(t)| > 1, \end{cases} \\
a_{33}(x_{i3}(t)) &= \begin{cases} -0.45, |x_{i3}(t)| \le 1, \\ 0.48, |x_{i3}(t)| > 1, \end{cases} \\
b_{11}(x_{i1}(t)) &= \begin{cases} 0.58, |x_{i1}(t)| \le 1, \\ -0.49, |x_{i1}(t)| > 1, \end{cases} \\
b_{12}(x_{i1}(t)) &= \begin{cases} -0.66, |x_{i1}(t)| \le 1, \\ -0.22, |x_{i1}(t)| > 1, \end{cases} \\
b_{13}(x_{i1}(t)) &= \begin{cases} 0.38, |x_{i1}(t)| \le 1, \\ 0.53, |x_{i1}(t)| > 1, \end{cases} \\
b_{21}(x_{i2}(t)) &= \begin{cases} -0.32, |x_{i2}(t)| \le 1, \\ 0.28, |x_{i2}(t)| > 1, \end{cases} \\
b_{22}(x_{i2}(t)) &= \begin{cases} 0.40, |x_{i2}(t)| \le 1, \\ 0.32, |x_{i2}(t)| > 1, \end{cases} \\
b_{23}(x_{i2}(t)) &= \begin{cases} -0.56, |x_{i2}(t)| \le 1, \\ -0.30, |x_{i2}(t)| > 1, \end{cases} \\
b_{31}(x_{i3}(t)) &= \begin{cases} 0.24, |x_{i3}(t)| \le 1, \\ -0.36, |x_{i3}(t)| > 1, \end{cases} \\
b_{32}(x_{i3}(t)) &= \begin{cases} 0.34, |x_{i3}(t)| \le 1, \\ 0.18, |x_{i3}(t)| > 1, \end{cases} \\
b_{33}(x_{i3}(t)) &= \begin{cases} 0.46, |x_{i3}(t)| \le 1, \\ -0.55, |x_{i3}(t)| > 1. \end{cases}
\end{aligned} \tag{39}$$

Obviously, $f_\rho(\cdot)$ as well as $g_\rho(\cdot)$ satisfy Assumption 1 with $\lambda_\rho = \tilde{\lambda}_\rho = 0.25$ and $\mu_\rho = \tilde{\mu}_\rho = 0.25$. Besides, $\mathbb{R}^3 \ni x^* = (0,0,0)^T$ is an equilibrium point of the isolated node for MWCMNN (37).

The output vector $y_i(t) \in \mathbb{R}^3$ is chosen as follows:

$$y_i(t) = Q_1 z_i(t) + Q_2 u_i(t), \quad i = 1, 2, \ldots, 6, \tag{40}$$

in which

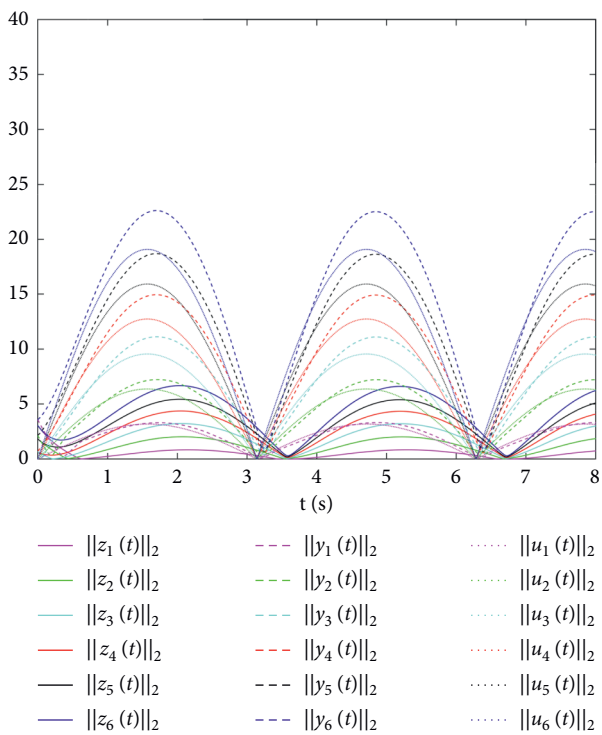


Figure 1: $\|z_i(t)_2\|, \|y_i(t)_2\|, \|u_i(t)\|_2$, $i = 1, 2, \ldots, 6$.

$$Q_1 = \begin{pmatrix} 0.4 & 0.3 & 0.2 \\ 0.1 & 0.5 & 0.3 \\ 0.6 & 0.4 & 0.6 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 0.3 & 0.4 \\ 0.2 & 0.5 \\ 0.6 & 0.1 \end{pmatrix}. \tag{41}$$

Case 1. The following matrix F which satisfies (17) can be derived with the assistance of the MATLAB YALMIP Toolbox:

$$F = I_6 \otimes \begin{pmatrix} -0.6168 & 8.0766 \\ -1.6951 & 0.2803 \\ 4.1206 & -3.0097 \end{pmatrix}. \tag{42}$$

In accordance with Theorem 1, the MWCMNN (37) is passive via the controller (13).

Case 2. The following matrices F and L_1 which satisfy (28) can be derived with the assistance of the MATLAB YALMIP Toolbox:

$$F = I_6 \otimes \begin{pmatrix} -0.2914 & 12.7012 \\ -2.1342 & 1.2881 \\ 4.8077 & -5.9296 \end{pmatrix}, \quad L_1 = I_6 \otimes \begin{pmatrix} 1.1852 & -0.0482 \\ -0.0482 & 2.5658 \end{pmatrix}. \tag{43}$$

By Theorem 2, the MWCMNN (37) is input-strictly passive via the controller (13).

Case 3. The following matrices F and L_2 which satisfy (32) can be derived with the assistance of the MATLAB YALMIP Toolbox:

$$F = I_6 \otimes \begin{pmatrix} -1.4129 & 6.4117 \\ -0.9625 & 1.8288 \\ 5.2249 & -1.8118 \end{pmatrix}, \quad L_2 = I_6 \otimes \begin{pmatrix} 2.2956 & -0.2844 & -0.6467 \\ -0.2844 & 2.4627 & -0.4603 \\ -0.6467 & -0.4603 & 1.8159 \end{pmatrix}. \tag{44}$$

On account of Theorem 3, the MWCMNN (37) is output-strictly passive via the controller (13). Figure 1 exhibits the simulation results.

5. Conclusion

It is the first time that the passivity of MWCMNNs has been discussed in this paper. Based on the Lyapunov stability theory, feedback control theory, and functional differential equations, several novel criteria have been set up to guarantee that the considered network is passive, output-strictly passive, and input-strictly passive. Finally, one numerical simulation example has been presented to demonstrate the effectiveness of the theoretical results. In the future work, the investigation of the synchronization and passivity for multiple weighted CMNNs with time-varying delays via adaptive control will be considered.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Scientific Research Foundation of Chongqing University of Technology under Grant 2020ZDZ028, Grant 2021ZDZ010, and Grant 2021ZDZ001.

References

[1] G. S. Moussa and M. Owais, "Modeling hot-mix asphalt dynamic modulus using deep residual neural networks: parametric and sensitivity analysis study," *Construction and Building Materials*, vol. 294, Article ID 123589, 2021.

[2] M. Owais, G. S. Moussa, and K. F. Hussain, "Robust deep learning architecture for traffic flow estimation from a subset of link sensors," *Journal of Transportation Engineering*, vol. 146, no. 1, Article ID 04019055, 2020.

[3] G. S. Moussa and M. Owais, "Pre-trained deep learning for hot-mix asphalt dynamic modulus prediction with laboratory

effort reduction," *Construction and Building Materials*, vol. 265, Article ID 120239, 2020.

[4] L. O. Chua and S. M. Sung Mo Kang, "Memristive devices and systems," *Proceedings of the IEEE*, vol. 64, no. 2, pp. 209–223, 1976.

[5] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[6] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[7] H. Bao, J. H. Park, and J. Cao, "Exponential synchronization of coupled stochastic memristor-based neural networks with time-varying probabilistic delay coupling and impulsive delay," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 1, pp. 190–201, 2016.

[8] Y. Fan, X. Huang, Y. Li, J. Xia, and G. Chen, "Aperiodically intermittent control for quasi-synchronization of delayed memristive neural networks: an interval matrix and matrix measure combined method," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2254–2265, 2019.

[9] J. Xiao and S. Zhong, "Synchronization and stability of delayed fractional-order memristive quaternion-valued neural networks with parameter uncertainties," *Neurocomputing*, vol. 363, pp. 321–338, 2019.

[10] M. Itoh and L. O. Chua, "Memristor cellular automata and memristor discrete-time cellular neural networks," *International Journal of Bifurcation and Chaos*, vol. 19, no. 11, pp. 3605–3656, 2009.

[11] S. Duan, X. Hu, Z. Dong, L. Wang, and P. Mazumder, "Memristor-based cellular nonlinear/neural network: design, analysis, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1202–1213, 2015.

[12] W. Zhang, C. Li, T. Huang, and J. Huang, "Stability and synchronization of memristor-based coupling neural networks with time-varying delays via intermittent control," *Neurocomputing*, vol. 173, pp. 1066–1072, 2016.

[13] P. Wan and J. Jian, "Passivity analysis of memristor-based impulsive inertial neural networks with time-varying delays," *ISA Transactions*, vol. 74, pp. 88–98, 2018.

[14] L. Zhou, "Delay-dependent and delay-independent passivity of a class of recurrent neural networks with impulse and multi-proportional delays," *Neurocomputing*, vol. 308, pp. 235–244, 2018.

[15] Y. Wang, Z. Yang, T. Liu, and H.-A. Tang, "Passivity and synchronization of multiple multi-delayed neural networks via impulsive control," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 6021687, 11 pages, 2020.

[16] Z. Meng and Z. Xiang, "Passivity analysis of memristor-based recurrent neural networks with mixed time-varying delays," *Neurocomputing*, vol. 165, pp. 270–279, 2015.

[17] J. Xiao, S. Zhong, and Y. Li, "New passivity criteria for memristive uncertain neural networks with leakage and time-varying delays," *ISA Transactions*, vol. 59, pp. 133–148, 2015.

[18] A. Wu and Z. Zeng, "Exponential passivity of memristive neural networks with time delays," *Neural Networks*, vol. 49, pp. 11–18, 2014.

[19] H.-A. Tang, S. Duan, X. Hu, and L. Wang, "Passivity and synchronization of coupled reaction-diffusion neural networks with multiple time-varying delays via impulsive control," *Neurocomputing*, vol. 318, pp. 30–42, 2018.

[20] H.-A. Tang, C.-X. Yue, and S. Duan, "Finite-time synchronization and passivity of multiple delayed coupled neural networks via impulsive control," *IEEE Access*, vol. 8, pp. 33532–33544, 2020.

[21] C.-X. Yue, L. Wang, X. Hu, H.-A. Tang, and S. Duan, "Pinning control for passivity and synchronization of coupled memristive reaction-diffusion neural networks with time-varying delay," *Neurocomputing*, vol. 381, pp. 113–129, 2020.

[22] W. Chen, Y. Zhang, and Y. Zheng, "Dissipativity of Markovian multiple-weighted coupled neural networks with dynamic event-triggered pinning control," *IET Control Theory & Applications*, vol. 14, no. 15, pp. 2030–2037, 2020.

[23] X. Li, W. Zhang, J.-A. Fang, and H. Li, "Event-triggered exponential synchronization for complex-valued memristive neural networks with time-varying delays," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 4104–4116, 2020.

[24] D. Lin, X. Chen, G. Yu, Z. Li, and Y. Xia, "Global exponential synchronization via nonlinear feedback control for delayed inertial memristor-based quaternion-valued neural networks with impulses," *Applied Mathematics and Computation*, vol. 401, Article ID 126093, 2021.

[25] L. Zhang, Y. Yang, and X. Xu, "Synchronization analysis for fractional order memristive Cohen-Grossberg neural networks with state feedback and impulsive control," *Physica A: Statistical Mechanics and its Applications*, vol. 506, pp. 644–660, 2018.

[26] J. C. Willems, "Dissipative dynamical systems part I: general theory," *Archive for Rational Mechanics and Analysis*, vol. 45, no. 5, pp. 321–351, 1972.

[27] X.-S. Ding, J.-D. Cao, and F. E. Alsaadi, "Passivity analysis of coupled inertial neural networks with time-varying delays and impulsive effects," *Pramana—Journal of Physics*, vol. 91, no. 5, p. 69, 2018.

[28] N. Li and J. Cao, "Passivity and robust synchronisation of switched interval coupled neural networks with time delay," *International Journal of Systems Science*, vol. 47, no. 12, pp. 2827–2836, 2016.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 7185830, 19 pages
https://doi.org/10.1155/2021/7185830

Research Article

Research on Importance Evaluation of Complex Product Parts Based on Multilayer Complex Network

Weiming Yang,[1] **Congdong Li**,[1] **Yinyun Yu**,[1] **and Mingsheng Zhong**[2]

[1]*School of Management, Jinan University, Guangzhou, Guangdong 510632, China*
[2]*Gree Electric Appliances, Inc.of Zhuhai, Zhuhai, Guangdong 519099, China*

Correspondence should be addressed to Congdong Li; licd@jnu.edu.cn

Received 26 May 2021; Revised 12 July 2021; Accepted 13 August 2021; Published 2 September 2021

Academic Editor: Jinliang Wang

The identification and evaluation of important parts play an important role in effective production arrangement and shortening the product development in the product design stage. In this paper, a complex product node importance evaluation method based on multilayer network is proposed to identify and evaluate importance parts faster. First, a complex product design expression network based on "function behavior structure (FBS)" multilayer complex network is established. Second, the evaluation index system of important design parts for complex products based on multilayer network is constructed. Third, a three-parameter grey relational model based on the fuzzy analytic hierarchy process and the Gini coefficient method is proposed. Finally, this method is available and feasible through taking the large permanent magnet synchronous centrifugal unit as an example.

1. Introduction

Complex product design process has the characteristics of diversity, nonlinearity, uncertainty, and evolution, which needs the efficient support and cooperation of various resources. Its architecture is highly complex and high dimensional. In this complex design process, on the one hand, companies have to invest a lot of manpower, material resources, and financial resources and require precise budgets and production arrangements. On the other hand, it is often interdisciplinary and accompanied by grey information. The experience and knowledge of experts in various fields are needed. Complex network can well represent the complex product design process. Many scholars use complex networks to realize the design control of complex products in recent years. Among them, the identification of important parts of complex products is of great value to improve the efficiency and sensitivity of the design [1, 2].

The identification of important nodes has attracted the attention of researchers and practitioners. Many diffusion models have been proposed to identify important nodes in complex networks [3–5]. The method commonly used to identify the most influential node set is the central node. Many methods can be used to measure the centrality of complex network structure, such as network degree, intermediate number, and tightness. However, measuring the centrality of the network structure to identify the nodes with maximum influence is not the best way to solve this problem. There are few studies on the evaluation of important nodes related to complex product engineering design. However, the research on the important nodes of complex products is of great significance for the design knowledge of control nodes. Therefore, this paper proposes a new important node identification method based on the multilayer network. Firstly, a multilevel design expression network considering the function, structure, and behavior of a complex product design is constructed. Then, a comprehensive new evaluation index system about network characteristics and component value is established. In addition, an improved three-parameter interval grey number model is used for evaluation considering the multidomain heterogeneity and difficult acquisition of knowledge. In the improvement, in order to better adapt to the research of this paper, we integrate the fuzzy analytic hierarchy process and the Gini coefficient method to improve the method.

The remainder of this paper is organized as follows. After reviewing some relevant literature in Section 2, we describe the research problem in Section 3 and give determination of evaluation index system in Section 4. In Section 5, we give the research methodology in this paper. In Section 6, we provide a case study about large permanent magnet synchronous centrifugal unit. Section 7 concludes this work.

2. Literature Review

Many scholars have proposed many methods for important node identification in recent years. Lu et al. [6] designed a cascade discount algorithm to solve the influence maximization problem. Gong et al. [7] proposed a discrete particle swarm optimization algorithm to optimize the local influence criterion. The representations and update rules for the particles are redefined in the proposed algorithm. Bao et al. [8] proposed a heuristic clustering (HC) algorithm based on the similarity index to classify nodes into different clusters, and finally, the center nodes in clusters are chosen as the multiple spreaders. HC algorithm not only ensures that the multiple spreaders are dispersively distributed in networks but also avoids the selected nodes to be very "negligible." Cui et al. [9] proposed a degree-descending search strategy (DDS) and developed an evolutionary algorithm that is capable of improving the efficiency significantly by eliminating the time-consuming simulations of the greedy algorithms. Li et al. [10] proposed an improved label propagation algorithm named LPA-MNI by combining the modularity function and node importance with the original LPA. Berberler et al. [11] conducted node importance analysis in wheel-related networks by a method of evaluating node importance by node contraction based on network agglomeration in communication networks. Sun et al. [12] proposed an entropy-based self-adaptive node importance evaluation method to evaluate node importance objectively.

Some of the above studies are aimed at pure networks, and most of them only study some characteristics of networks. It is difficult to reflect the importance of nodes in an all-round way and cannot effectively identify the important nodes in practical application. We start with multilayer network and establish a new index. This index system includes the value of parts in the design, which is more comprehensive.

There are many research studies on the description network of complex products, mainly focusing on the research of single-layer and weighted complex networks. Bencherif and Mouss [13] proposed a product development process model in an innovation context and strategy framework of design process and project management. The process modelling is based on complex network theory, to improve characterization analysis for product development process modelling. Li et al. [14] focused on the impacts of organization-component executing patterns on the risk propagation in CPD interdependent networks considering the limited risk resisting capacity of organizations. Li et al. [15] developed a general model based on complex network to depict the dynamic design change propagation. Numerical simulations are conducted to explore the general law of the propagation and investigate the influences of design change tolerance capacity distribution (alpha, beta), attack strategies, and recovery capacity (gamma). Li et al. [16], for the first time, applied the SoV into the research on quality risk propagation of complex product collaborative manufacturing supply chain network. Yin et al. [17] proposed a method for identifying the influential parts of a CMP based on complex network theory from the perspective of reliability. It is used to identify the influential parts in each community of products. Yu et al. [18] built a directed weighted complex product network model to represent the product structure under given requirement by defining interconnections among parts. Zhang et al. [19] proposed a novel HEO modelling method based on complex networks' theory. Zhang et al. [20] proposed a new four-phase routing approach based on weighted and directed complex networks for multisource design change propagation.

It can be seen from the existing design networks that single-layer networks are often used to study their properties. However, some design factors will be ignored. So, this paper uses a comprehensive design process description model, FBS model, to describe the design process. The comprehensive evaluation index system is established. Then, the three-parameter interval grey number grey relational model is used to evaluate the important nodes.

There are many studies on three-parameter interval grey number relational model. Considering the value information of the three-parameter interval grey number and the risk attitude of decision maker, Zhang et al. [21] advanced a multiattribute group grey target decision-making method based on a three-parameter interval grey number. From the perspective of three-way decision space, Li and Zhang [22] combined the theories of interval concept lattice and three-way decision and then put forward interval three-way decision space theory. To solve decision-making problems having the characteristics of grey systems, Li et al. [23] introduced the three-parameter interval grey number to measure the evaluation index. Chen and Chen [24] proposed a dynamic multiattribute decision-making method based on the prospect theory for dealing with the dynamic multiattribute decision-making problem with the three-parameter interval grey number. Aiming at the multiple attribute decision-making problem with three-parameter interval grey numbers, Li and Zhang [25] proposed a grey-incidence clustering decision-making method based on regret theory. In consideration of the fuzziness and the uncertainty in decision information, Li et al. [26] proposed a three-parameter interval grey linguistic variable decision-making method based on projection model and prospect theory. It can be seen from the previous literature that scholars have carried out a lot of research studies on three-parameter interval grey number relational model. However, there are still some areas to be studied about the determination of its weight. In order to make it more suitable for this paper, this study comprehensively considers its weight setting.

To sum up, we can see that the single-layer network is mostly used to analyze the network center from the expression of the design process. The research on the identification method of important network nodes mainly adopts

the search algorithm. In the process of subcomplex product design, the identification of important nodes is very important, which determines the speed and quality of the following links. Firstly, this paper establishes a multilayer network model considering the functional behavior structure knowledge in the process of the product design. Then, a new evaluation index is determined for the network model, including the network characteristics and the value of parts. We improve the three-parameter interval grey number relational model to adapt to the research. Finally, we use the three-parameter grey relational model based on TFAHP and Gini coefficient method to evaluate and rank the important nodes in engineering change. The research framework is shown in Figure 1.

3. Problem Description

Previous studies have shown that the FBS model is a more comprehensive description model of the design process, in which the function represents what the design object is used to do, behavior describes what the design object does, and structure specifies what the design object is. In the process of complex product design, the FBS model can be used to formalize the parameters of the parts according to the new function, behavior, and structure requirements. In addition, Hamraz et al. also emphasized that the use of the FBS model to express the relationships in complex products can effectively improve the product design. Therefore, this paper considers the FBS relationship to formalize the complex product design process.

The realization of function is based on the exchange of material, information, and energy between the product and the external environment. When two parts share the functions in the form of material flow, information flow, and energy flow, we define that there is a functional relationship between the parts. Product function is an abstract concept, which can make the function relationship have strong concealment and often cannot be perceptual cognition by designers.

Behavior is a bridge between the function and the structure and an objective description of function realization. It contains three independent implicit behavior attributes: mechanical (strength, inertia, elasticity, etc.), electrical characteristics (conduction, resistance, charging, etc.), and thermal effects (conduction, temperature change, absorption, etc.). Due to the behavior requirements of products, the combination of behavior attributes can be formed between parts with the same behavior attributes.

Structure is the physical carrier of function, which emphasizes the specific composition of the physical entity to realize the function. In particular, structural attributes are explicit attributes used to describe physical entities, which mainly include geometry (size, shape, etc.), material (type, volume, density, etc.), surface (surface roughness, texture, etc.), and controller (microchip, relay, etc.). Similarly, due to the structural requirements of products, the combination of structural attributes exists between components with the same structural attributes (Figure 2). The multilayer network construction process is shown in Figure 2.

The n components of a complex product and their associated relationships are represented as a single-layer subnetwork $G\alpha = (V, E\alpha, W\alpha)$, where $\alpha = 1, 2, 3$ are the network levels. It represents the network level and represents the single-layer subnetwork type (functional network, behavioral network, and structural network). $V = (vi|\, i = 1, 2, \ldots, N)$ is the set of network nodes. It represents the components set. $E\alpha = \{ei\ \alpha, j \mid i = 1, 2, \ldots, N; j = 1, 2, \ldots, N; i \neq j\}$ is the set of connected edges of the network. It represents the association set. $W\alpha = \{wi\,\alpha, j \mid i = 1, 2, \ldots, N; j = 1, 2, \ldots, N; i \neq j\}$ is the weight set of connected edges. It represents the association strength set. For node vi and node vj in the network, if there is an association relationship between the corresponding components, then there is a connected edge $ei\ \alpha, j = 1$.

4. Determination of Evaluation Index System

Compared with the traditional complex network, the multinetwork model can better reflect the diversity of association relations in complex products and the rich topology characteristics in complex products. In order to evaluate the influence of each node, we need to comprehensively consider the local attributes, global attributes, and their specific significance in complex products from multiple perspectives. At the same time, we consider the indicators that can reflect the properties of parts of complex products to form a comprehensive evaluation index. The detailed indicators are as follows:

(1) Clustering coefficient: it refers to the parameters used to measure the clustering of network nodes. Intuitively speaking, clustering coefficient describes the possibility that a node's friends are still friends. The higher the clustering coefficient of a node, the more likely it is that its friend is still a friend, and the more important it is in the whole network circle. Generally, assuming that a node i in an undirected network has k adjacent nodes, then the ratio of the actual number of edges to the possible number of edges among the k nodes is called the clustering coefficient: $C_i = (E_i/(k(k-1)/2)) * k$, where k is the degree of node i, which is equivalent to counting the number of friends of node i and E_i is the actual number of edges between k nodes connected with node i.Suppose there are n nodes in a complex network; then, the Erdos number of each node is defined as follows: $D_i = (1/(N-1))\sum_{i,j\neq i} d_{ij}$, where d_{ij} is the length of the shortest path from node i to node j.

(2) Erdos: assuming that the current node is Erdos and the Erdos number of other nodes is the length of the shortest path from the current node to other nodes, then the Erdos number of the current node is the length of the average path from the current node to other nodes.

(3) Node betweenness: the number of paths that all shortest paths in the network pass through this node. The betweenness B_i of vertex i is defined as

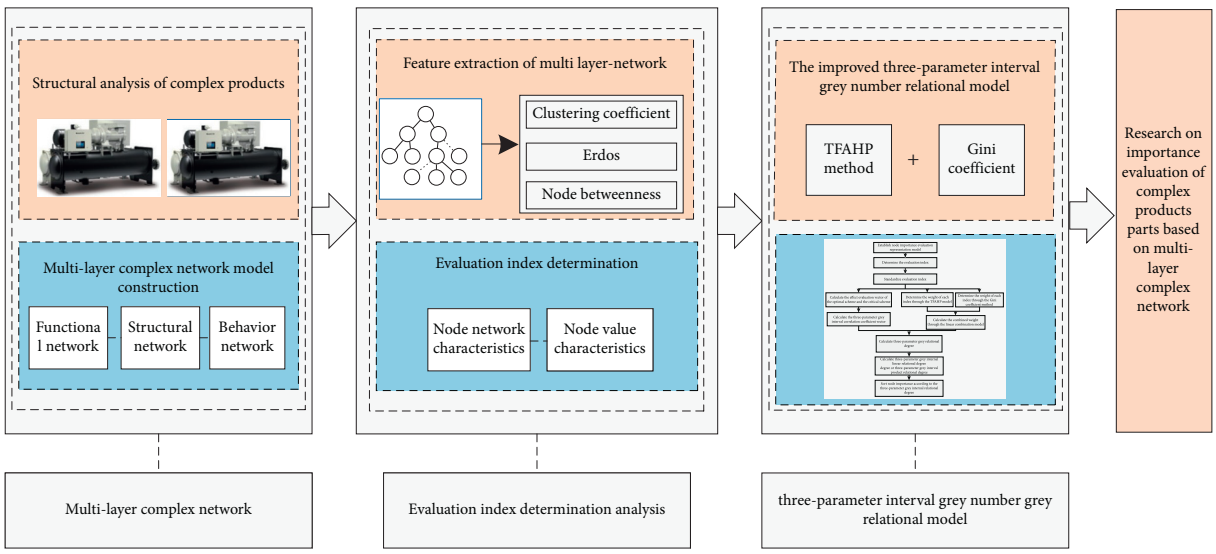


Figure 1: Framework of the proposed complex product node importance evaluation method.

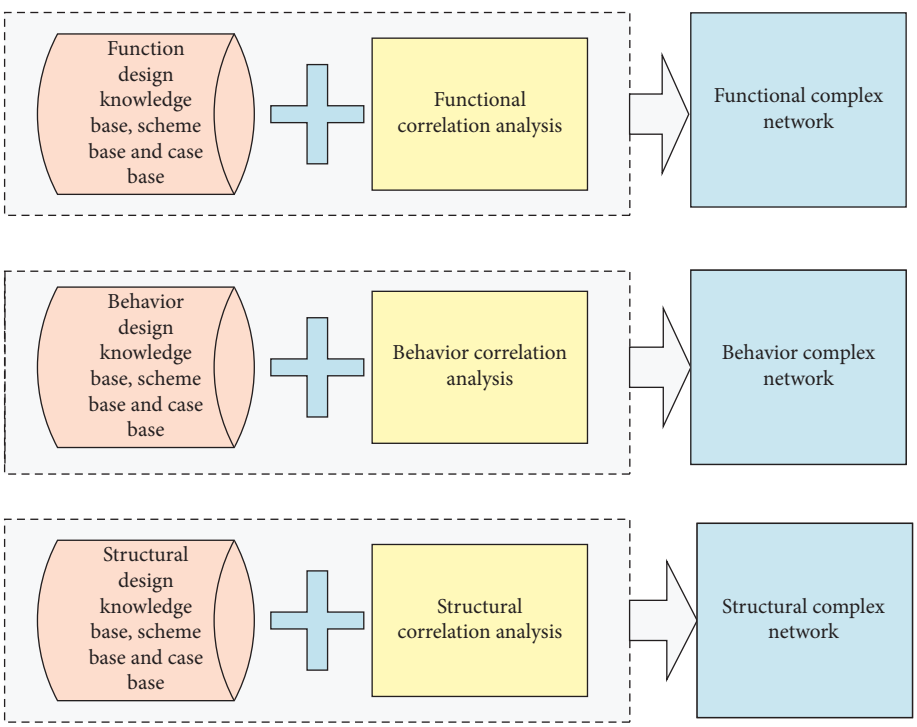


Figure 2: Multilayer complex network analysis of complex product design knowledge.

$B_i = \sum_{j,k\in V}(n_{jk}(i)/n_{jk})$, where $n_{jk}(i)$ is the number of shortest paths between nodes j and k.

(4) Parts value: the engineering change of different parts needs different change costs. And, different nodes have different values. Therefore, the node's own attribute is the value of node which is used as the evaluation index of node importance. It can be expressed as $C_{(v_i)} = c_{(v_m)} + c_{(v_l)} + c_{(v_e)} + c_{(v_o)}$, where c_i is the total value of node i and $C_{(v_m)}$ is the material cost if node i. It refers to the cost of product standard consumption, supporting raw materials, product accessories, and various materials used for production or providing services. It mainly includes the purchase price, related taxes, freight, loading and unloading fees, insurance premiums, and other costs that can be directly attributable to the acquisition of materials. $C_{(v_l)}$ is the labor cost. It refers to the remuneration and other expenses paid to employees in order to obtain the services provided by employees. It mainly includes the salary, bonus, allowance,

welfare, and education fund. $C_{(v_m)}$ is the manufacturing cost. It refers to energy consumption, manufacturing accessories, labor insurance, and office and fixed expenses. $C_{(v_o)}$ is the other costs, some consumption including fuel cost, power cost, office cost, and depreciation consumed by each production unit.

5. Research Methodology

5.1. Three-Parameter Interval Grey Number. From the definition of three-parameter interval grey number, it can be known that it refers to the interval grey number where the center of gravity point with the greatest possible value is known. It can be marked as $A(\otimes) = [\underline{a}, \tilde{a}, \overline{a}]$, where $\underline{a} \le \tilde{a} \le \overline{a}$, and $\underline{a}$ and $\overline{a}$ are the upper and lower limits of the interval, respectively. $\tilde{a}$ is called the "center of gravity" point ([27, 28]).

When two of the three parameters $\underline{a}$, $\tilde{a}$, and $\overline{a}$ are the same, the three-parameter interval grey number degenerates to the interval grey number. When $\underline{a} = \tilde{a} = \overline{a}$, the three-parameter interval grey number degenerates to the real number. In fact, the interval grey number and the real number are special cases of the three-parameter interval grey number.

Its algorithm is similar to the interval grey number. Let three-parameter interval grey number be $A(\otimes) = [\underline{a}, \tilde{a}, \overline{a}]$ and $B(\otimes) = [\underline{b}, \tilde{b}, \overline{b}]$; then,

(i) $A(\otimes) + B(\otimes) = \left\lfloor \underline{a} + \underline{b}, \tilde{a} + \tilde{b}, \overline{a} + \overline{b} \right\rfloor$

(ii) $(A(\otimes)/B(\otimes)) \in [\min\{(\underline{a}/\underline{b}), (\underline{a}/\overline{b}), (\overline{a}/\underline{b}), (\overline{a}/\overline{b})\}, (\tilde{a}/\tilde{b}), \max\{(\underline{a}/\underline{b}), (\underline{a}/\overline{b}), (\overline{a}/\underline{b}), (\overline{a}/\overline{b})\}]$

(iii) $\lambda A(\otimes) = [\underline{\lambda a}, \lambda\tilde{a}, \lambda\overline{a}]$, $\lambda > 0$

5.2. Three-Parameter Interval Grey Number Grey Relational Model. Suppose that there are n alternative engineering change schemes. They constituted by evaluation schemes' set $A = \{a_1, a_2, \ldots, a_n\}$. The index set $S = \{s_1, s_2, \ldots, s_m\}$ is composed of m attributes. The index value of scheme a_i under the evaluation index s_j can be expressed as $u_{ij}(\otimes) = \left\lfloor \underline{u}_{ij}, \tilde{u}_{ij}, \overline{u}_{ij} \right\rfloor$ $(\underline{u}_{ij} \le \tilde{u}_{ij} \le \overline{u}_{ij}, i = 1, 2, \ldots, n; j = 1, 2, \ldots, m)$. The effect evaluation vector of each scheme is $u_i(\otimes) = (u_{i1}(\otimes), u_{i2}(\otimes), \ldots, u_{im}(\otimes))$, $i = 1, 2, \ldots, n$. The weight of index under each scheme is $w_{i1}, w_{i2}, \ldots, w_{im}$, and $\sum_{j=1}^{m} w_{ij} = 1$ $(i = 1, 2, \ldots, n)$. There are different attribute indexes with different dimensions and measurement standards. In order to increase the comparability of alternatives, it is necessary to normalize the effect evaluation vector of decision alternatives. In this paper, we use the range transformation method to normalize the decision matrix.

For profitable attribute values,

$$\underline{x}_{ij} = \frac{\underline{u}_{ij} - \underline{u}_j^{\nabla}}{\overline{u}_j^* - \underline{u}_j^{\nabla}}, \quad \tilde{x}_{ij} = \frac{\tilde{u}_{ij} - \underline{u}_j^{\nabla}}{\overline{u}_j^* - \underline{u}_j^{\nabla}}, \quad \overline{x}_{ij} = \frac{\overline{u}_{ij} - \underline{u}_j^{\nabla}}{\overline{u}_j^* - \underline{u}_j^{\nabla}}. \tag{1}$$

For cost attribute values,

$$\underline{x}_{ij} = \frac{\overline{u}_j^* - \overline{u}_{ij}}{\overline{u}_j^* - \underline{u}_j^{\nabla}}, \quad \tilde{x}_{ij} = \frac{\overline{u}_j^* - \tilde{u}_{ij}}{\overline{u}_j^* - \underline{u}_j^{\nabla}}, \quad \overline{x}_{ij} = \frac{\overline{u}_j^* - \overline{u}_{ij}}{\overline{u}_j^* - \underline{u}_j^{\nabla}}, \tag{2}$$

where $\overline{u}_j^* = \max_{1 \le i \le n}\{\overline{u}_{ij}\}, \underline{u}_j^{\nabla} \min_{1 \le i \le n}\{\underline{u}_{ij}\}$, $j = 1, 2, \ldots, m$.

Let the normalized effect evaluation vector be

$$x_i(\otimes) = (x_{i1}(\otimes), x_{i2}(\otimes), \ldots, x_{im}(\otimes)), \quad i = 1, 2, \ldots, n, \tag{3}$$

where $x_{ij}(\otimes) \in [\underline{x}_{ij}, \tilde{x}_{ij}, \overline{x}_{ij}]$ is a three-parameter interval grey number in $[0, 1]$.

It is recorded that $\underline{x}_j^+ = \max_{1 \le i \le n}\{\underline{x}_{ij}\}$, $\tilde{x}_j^+ = \max_{1 \le i \le n}\{\tilde{x}_{ij}\}$, $\overline{x}_j^+ = \max_{1 \le i \le n}\{\overline{x}_{ij}\}$, $\underline{x}_j^- = \min_{1 \le i \le n}\{\underline{x}_{ij}\}$, $\tilde{x}_j^- = \min_{1 \le i \le n}\{\tilde{x}_{ij}\}$, and $\overline{x}_j^- = \min_{1 \le i \le n}\{\overline{x}_{ij}\}$ $(j = 1, 2, \ldots, m)$. Then, the m-dimensional three-parameter non-negative interval grey number vectors,

$$x^+(\otimes) = \{x_1^+(\otimes), x_2^+(\otimes), \ldots, x_m^+(\otimes)\}, \quad x^-(\otimes) = \{x_1^-(\otimes), x_2^-(\otimes), \ldots, x_m^-(\otimes)\}, \tag{4}$$

are called ideal optimal scheme effect evaluation vectors and critical scheme effect evaluation vectors, respectively [29].

We assume that the grey interval relational degree of the normalized effect evaluation vector $x_i(\otimes)$ of scheme A_i with respect to the ideal optimal scheme effect evaluation vector $x^+(\otimes)$ is $G(x^+(\otimes), x_i(\otimes))$. And, the grey interval relational degree of the critical scheme effect evaluation vector $x^-(\otimes)$ is $G(x^-(\otimes), x_i(\otimes))$. Assume that the weights of two grey relational degrees are α_1, α_2 $(\alpha_1 + \alpha_2 = 1)$. Then,

$$G(x_i(\otimes)) = \alpha_1 G(x^+(\otimes), x_i(\otimes)) + \alpha_2 [1 - G(x^-(\otimes), x_i(\otimes))], \quad i = 1, 2, \ldots, n, \tag{5}$$

is the three-parameter grey interval linear relational degree of the effect evaluation vector $x_i(\otimes)$.

$$G(x_i(\otimes)) = [G(x^+(\otimes), x_i(\otimes))]^{\alpha_1} + [1 - G(x^-(\otimes), x_i(\otimes))]^{\alpha_2}, \quad i = 1, 2, \ldots, n, \tag{6}$$

is the three-parameter grey interval product relational degree of the effect evaluation vector $x_i(\otimes)$ (according to [28] and experts' opinion, this paper is set to $\lambda = 0.5$, $\xi = 0.5$, and $\beta_1 = \beta_2 = 0.5$).

The distribution probability of barycenter point with the highest probability of taking the value of three-parameter interval grey number $x_{ij}(\otimes) \in [\underline{x}_{ij}, \tilde{x}_{ij}, \overline{x}_{ij}]$ is $f(\tilde{x}_{ij}) \geq \sigma$. Normally, $\sigma \geq 60\%$. If $\sigma \leq 60\%$, it indicates that the decision is wrong, and the most likely value needs to be determined again. Based on the center of gravity, we can build a three-parameter interval grey number relational degree evaluation model.

Definition 3. For the three-parameter interval grey number $x_{ij}(\otimes) \in [\underline{x}_{ij}, \tilde{x}_{ij}, \overline{x}_{ij}]$,

$$\gamma_{ij}^+ = \frac{3}{5} \times \frac{\tilde{m}^+ + \eta\tilde{M}^+}{\tilde{\Delta}_{ij}^+ + \eta\tilde{M}^+} + \frac{2}{5}\left[(1-\beta)\frac{\underline{m}^+ + \eta\underline{M}^+}{\underline{\Delta}_{ij}^+ + \eta\underline{M}^+} \beta \frac{\overline{m}^+ + \eta\overline{M}^+}{\overline{\Delta}_{ij}^+ + \eta\overline{M}^+}\right], \tag{7}$$

is called the three-parameter grey interval relational coefficient of subfactor x_{ij} with respect to ideal factorx_j^+. $\eta \in (0,1)$ is the resolution coefficient. $\beta \in (0,1)$ is the decision preference coefficient (according to [28] and experts' opinion, normally, $\eta = \beta = 0.5$), where

$$\begin{aligned}
\underline{\Delta}_{ij}^+ &= \left|\underline{x}_j^+ - \underline{x}_{ij}\right|, \\
\tilde{\Delta}_{ij}^+ &= \left|\tilde{x}_j^+ - \tilde{x}_{ij}\right|, \quad i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, m, \\
\overline{\Delta}_{ij}^+ &= \left|\overline{x}_j^+ - \overline{x}_{ij}\right|, \\
\underline{m}^+ &= \min_{1\leq i\leq n} \min_{1\leq j\leq m} \underline{\Delta}_{ij}^+, \\
\tilde{m}^+ &= \min_{1\leq i\leq n} \min_{1\leq j\leq m} \tilde{\Delta}_{ij}^+, \\
\overline{m}^+ &= \min_{1\leq i\leq n} \min_{1\leq j\leq m} \overline{\Delta}_{ij}^+, \\
\underline{M}^+ &= \max_{1\leq i\leq n} \max_{1\leq j\leq m} \underline{\Delta}_{ij}^+, \\
\tilde{M}^+ &= \max_{1\leq i\leq n} \max_{1\leq j\leq m} \tilde{\Delta}_{ij}^+, \\
\overline{M}^+ &= \max_{1\leq i\leq n} \max_{1\leq j\leq m} \overline{\Delta}_{ij}^+, \\
G(x^+(\otimes), x_i(\otimes)) &= \sum_{j=1}^{m} w_{ij}\gamma_{ij}^+, \quad i = 1, 2, \ldots, n,
\end{aligned} \tag{8}$$

is called the three-parameter grey interval relational degree of the effect evaluation vector $x_i(\otimes)$ about the ideal optimal scheme effect evaluation vector $x^+(\otimes)$.

Definition 4. For three-parameter interval grey number $x_{ij}(\otimes) \in [\underline{x}_{ij}, \tilde{x}_{ij}, \overline{x}_{ij}]$,

$$\gamma_{ij}^- = \frac{3}{5} \times \frac{\tilde{m}^- + \varepsilon\tilde{M}^-}{\tilde{\Delta}_{ij}^- + \varepsilon\tilde{M}^-} + \frac{2}{5} \times \left[(1-\delta)\frac{\underline{m}^- + \varepsilon\underline{M}^-}{\underline{\Delta}_{ij}^- + \varepsilon\underline{M}^-} + \delta\frac{\overline{m}^- + \varepsilon\overline{M}^-}{\overline{\Delta}_{ij}^- + \varepsilon\overline{M}^-}\right], \tag{9}$$

is called the three-parameter grey interval relational coefficient of sub factor x_{ij} with respect to ideal facto x_j^-. $\eta \in (0,1)$ is the resolution coefficient. $\delta \in (0,1)$ is the decision preference coefficient (according to [28] and experts' opinion, normally, $\varepsilon = \beta = 0.5$), where

$$\begin{aligned}
\underline{\Delta}_{ij}^- &= \left|\underline{x}_{ij} - \underline{x}_j^-\right|, \\
\tilde{\Delta}_{ij}^- &= \left|\tilde{x}_{ij} - \tilde{x}_j^-\right|, \quad i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, m, \\
\overline{\Delta}_{ij}^- &= \left|\overline{x}_{ij} - \overline{x}_j^-\right|, \\
\underline{m}^- &= \min_{1\leq i\leq n} \min_{1\leq j\leq m} \underline{\Delta}_{ij}^-, \\
\tilde{m}^- &= \min_{1\leq i\leq n} \min_{1\leq j\leq m} \tilde{\Delta}_{ij}^-, \\
\overline{m}^- &= \min_{1\leq i\leq n} \min_{1\leq j\leq m} \overline{\Delta}_{ij}^-, \\
\underline{M}^- &= \max_{1\leq i\leq n} \max_{1\leq j\leq m} \underline{\Delta}_{ij}^-, \\
\tilde{M}^- &= \max_{1\leq i\leq n} \max_{1\leq j\leq m} \tilde{\Delta}_{ij}^-, \\
\overline{M}^- &= \max_{1\leq i\leq n} \max_{1\leq j\leq m} \overline{\Delta}_{ij}^-, \\
G(x^-(\otimes), x_i(\otimes)) &= \sum_{j=1}^{m} w_{ij}\gamma_{ij}^-, \quad i = 1, 2, \ldots, n,
\end{aligned} \tag{10}$$

is called the three-parameter grey interval relational degree of the effect evaluation vector $x_i(\otimes)$ about the critical scheme effect evaluation vector $x^-(\otimes)$.

5.3. Determination of Weight. In order to better reflect the balance and scientificity of expert evaluation and objective evaluation in the evaluation process, this paper adopts a combined weight method. Firstly, the triangular fuzzy analytical hierarchy process method is used to determine the subjective weight. Then the Gini coefficient method is used to determine the objective weight. Finally, the linear weighting method is used to combine the two methods, so as to increase its scientificity.

5.3.1. Determination of Weight Based on Triangular Fuzzy Analytic Hierarchy Process (TFAHP). Analytic hierarchy process (AHP) is a multicriteria decision analysis method

which transforms subjective judgment into objective one. It makes people's subjective judgment process hierarchical and mathematical and provides a simple decision-making method for multiattribute decision-making problems.

Triangular fuzzy number has obvious advantages in solving the problems of complexity, fuzziness, and uncertainty. Fuzzy numbers have been studied extensively. [30–36] TFAHP (triangular fuzzy analytic hierarchy process) is a combination of fuzzy theory and AHP. It fully considers the subjective judgment of the evaluator, the fuzziness of decision, and the preference, which makes the decision result more objective and reasonable. The calculation steps are as follows:

(1) According to the above evaluation index system construction, the hierarchical model is established, as shown in Table 1. The top layer is target layer A. The middle layer is criterion layer B. The lowest layer is index layer C.

(2) Each expert can get the triangular fuzzy number judgment matrix of the target layer to the criterion layer and the criterion layer to the index layer by comparing each element. Let $\tilde{x}_{ij}^t$ be the triangular fuzzy number judgment matrix of the expert t.Then, the triangular fuzzy number is $\tilde{x}_{ij}^t = (\tilde{l}_{ij}^t, \tilde{m}_{ij}^t, \tilde{u}_{ij}^t)$, where $\tilde{l}_{ij}^t$ and $\tilde{u}_{ij}^t$ are the upper and lower bounds of fuzzy numbers and $\tilde{m}_{ij}^t$ is the fuzzy median. The smaller the value of $\tilde{u}_{ij}^t - \tilde{l}_{ij}^t$, the more fuzzy the judgment. When $\tilde{u}_{ij}^t - \tilde{l}_{ij}^t$ is 0, the judgment is not fuzzy.

Each element value of the fuzzy judgment matrix represents the importance of one factor relative to another. Because it is difficult to construct the consistency matrix by 1–9 and its reciprocal scaling method, the ranking result is far from the actual thinking 0.1–0.9 scale method which can better reflect the actual estimation of experts. Therefore, the 0.1–0.9 scale method is used in this study, as shown in Table 2.

(3) The judgment matrix of each expert is synthesized, and the triangular fuzzy number in the judgment matrix is changed into nonfuzzy number. Suppose the total number of experts is T and the weight of each expert is q_t; then, the synthetic judgment matrix is $\tilde{X} = \sum_{t=1}^{T} q_t \tilde{X}^t$.

According to the method of reference 35, the most likely estimated value of each factor in fuzzy judgment matrix A by pairwise comparison are extracted. The fuzzy complementary judgment matrix M: $M = \sum_{t=1}^{T} m_{ij}^t q_t$ is obtained. It can be transformed into fuzzy consistency matrix M' through formula $m_i' = \sum_{k=1}^{n} m_{ik}$ and $m_i' = ((m_i' - m_j')/2(n-1)) + 0.5$. If M' satisfies the requirement of consistency, then $\tilde{X}$ also satisfies the requirement of consistency. For the fuzzy consistent matrix M' of 0.1–0.9 scale, the consistency is checked by calculating δ and σ to judge whether the fuzzy complementary judgment matrix is consistent with the actual situation:

$$\delta = \max\left\{\left|m_{ij}' - m_{ij}\right|\right\},$$
$$\sigma = \frac{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(m_{ij}' - m_{ij}\right)^2}}{n}. \tag{11}$$

$\delta < 0.2$ and $\sigma < 0.1$, and the fuzzy matrix passes the consistency test, which is in line with the actual situation. Otherwise, the judgment matrix should be adjusted.

The fuzzy judgment matrix $\tilde{X}$ is transformed into nonfuzzy matrix F'':

$$f'' = \frac{l_{ij}}{2\left(l_{ij}+2\right)} + \frac{2m_{ij}}{3} + \frac{u_{ij}}{6}. \tag{12}$$

(4) In order to determine the index weight of each factor, the comprehensive importance of each element is calculated according to each fuzzy judgment matrix. It can be solved according to the weight formula:

$$\omega_i = \frac{\sum_{j=1}^{n} f_{ij}'' - 0.5}{\sum\left(\sum_{j=1}^{n} f_{ij}'' - 0.5\right)}. \tag{13}$$

5.3.2. Weight Determination Method Based on Gini Coefficient

(1) Principle of Gini Coefficient Weighting Method. The Gini coefficient weighting method is an objective weighting method by calculating Gini coefficient of evaluation index and normalizing Gini coefficient of each index. First of all, the different data of n evaluation objects of a specific evaluation index can be regarded as the income of different level people. Then, the Gini coefficient of a certain index can be calculated. The value of Gini coefficient can reflect the data difference between different evaluation objects. Then, in order to ensure that weight of all indexes are in the range of 0 to 1 and the sum is 1, the Gini coefficient value of each index will be normalized to get the Gini coefficient weight of the evaluation index [37].

(2) Gini Coefficient Weight Calculation of Evaluation Index. We assume that G_k is the Gini coefficient of the kth index, $Y_{ki}$is the ith data of the kth index, and μK is the expected value of all data of the kth index. Then, the Gini coefficient G_k of the kth index is shown as follows:

$$G_k = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\left|Y_{ki} - Y_{kj}\right|}{2n^2\mu_k}, \tag{14}$$

$$G_k = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\left|Y_{ki} - Y_{kj}\right|}{\left(n^2 - n\right)}. \tag{15}$$

Especially, when the mean value of index data is not 0, the Gini coefficient is calculated by the improved formula (14). When the mean value of the index data is 0, the Gini

Table 1: Index system hierarchical model of important nodes in the complex product design.

A	Node importance evaluation			
B	Link degree (B_1)	Average path length (B_2)	Shortest route number (B_3)	Value attributes of complex product (B_4)
C	Clustering coefficient (C_1)	Erdos (C_2)	Node betweenness (C_3)	Value of parts (C_4)

Table 2: 0.1–0.9 quantitative scale.

Scale	Meaning
0.1	When one factor is compared with another, the latter is extremely important
0.2	When one factor is compared with another, the latter is strongly important
0.3	When one factor is compared with another, the latter is obviously important
0.4	When one factor is compared with another, the latter is slightly important
0.5	When one factor is compared with another, the two factors are of equal importance
0.6	When one factor is compared with another, the former is slightly more important
0.7	When one factor is compared with another, the former is obviously important
0.8	When one factor is compared with another, the former is strongly important
0.9	When one factor is compared with another, the former is extremely important

coefficient of the index is calculated by the original formula (15). The Gini coefficient of the index truly reflects the data changes of different evaluation objects of the index.

Gini coefficient weight g_k of the kth index can be obtained by normalizing the Gini coefficient value of each index:

$$g_k = \frac{G_k}{\sum_{i=1}^{m} G_i}, \tag{16}$$

where g_k is Gini coefficient weight of the kth index, G_k is Gini coefficient value of the kth index, and m is the number of indexes.

The advantages of the Gini coefficient weighting method are as follows. First, the weight calculation is not affected by the unit dimension of the index, and the definition of Gini coefficient itself eliminates the dimensional influence. Second, the Gini coefficient value of the evaluation index reflects the difference between any two evaluation objects. Gini coefficient weight reflects the difference between the data of different evaluation objects of an index. And, the weight reflects the data information of the index, which meets the requirements of the objective weighting method.

5.3.3. Combination Weighting Method Based on TFAHP-Gini Coefficient. In order to be more scientific, we adopt the linear weighting method to combine the two methods after the weight determining based on subjective and objective.

The weight vector of linear combination is calculated and proved as follows.

Suppose N kinds of subjective and objective methods to get l kinds of weights. The weight vectors are $W_1, W_2, \ldots, W_L$. The kth weight direction quantity is $W^k = (\varpi_1^k, \varpi_2^k, \ldots, \varpi_m^k)$. It satisfies that $\sum_{j=1}^{m} \varpi_j^k = 1$, $\varpi_j^k \geq 0\,(j = 1, 2, \ldots, m; k = 1, 2, \ldots, L)$. The vector obtained by linear combination of multiple weight vectors is called linear combination weight vector. Let $W^* = (\varpi_1^*, \varpi_2^*, \ldots, \varpi_m^*)^T$ be the linear combination weight vector of L weight vectors. It can be expressed in the vector form as $W^* = \sum_{k=1}^{L} x_k w^k$, where x_k is the linear combination coefficient. It satisfies $\sum_{k=1}^{L} x_k = 1$, $x_k \geq 0$. The solution of linear combined weight vector is a multiobjective optimization problem. On the one hand, the sum of weighted generalized distances between all schemes and ideal schemes should be minimized. On the other hand, the uncertainty of combination coefficient should be eliminated as far as possible. According to the Jaynes maximum entropy principle, the comprehensive weight coefficient of the index should be determined so that the Shannon entropy takes the maximum value. Therefore, the following optimization decision model is established:

$$\begin{aligned} &\min\left[\mu \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{L} x_k \varpi_j^k \left(1 - r_{ij}\right) + (1-\mu)\sum x_k \ln x_k\right], \\ &\text{s.t.} \quad \sum_{k=1}^{L} x_k = 1, \quad x_k \geq 0, \end{aligned} \tag{17}$$

where r_{ij} is the relational coefficient, $0 \leq \mu \leq 1$. It can be used to express the balance coefficient between two targets, which can be given in advance according to practical problems.

According to [3], there is a unique solution for the single objective optimization problem (SP). The solution is as follows:

$$x = \left[\frac{s_1}{\sum_{k=1}^{l} s_k}, \frac{s_2}{\sum_{k=1}^{l} s_k}, \ldots, \frac{s_L}{\sum_{k=1}^{L} s_k}\right], \tag{18}$$

where $s_k = \exp\left\{-\left[1 + \mu \sum_{i=1}^{n}\sum_{j=1}^{m} \left(\varpi_j^k (1 - r_{ij})/(1-\mu)\right)\right\}$ $(k = 1, 2, \ldots, L)$, N is the number of evaluation schemes, and M is the number of evaluation indexes.

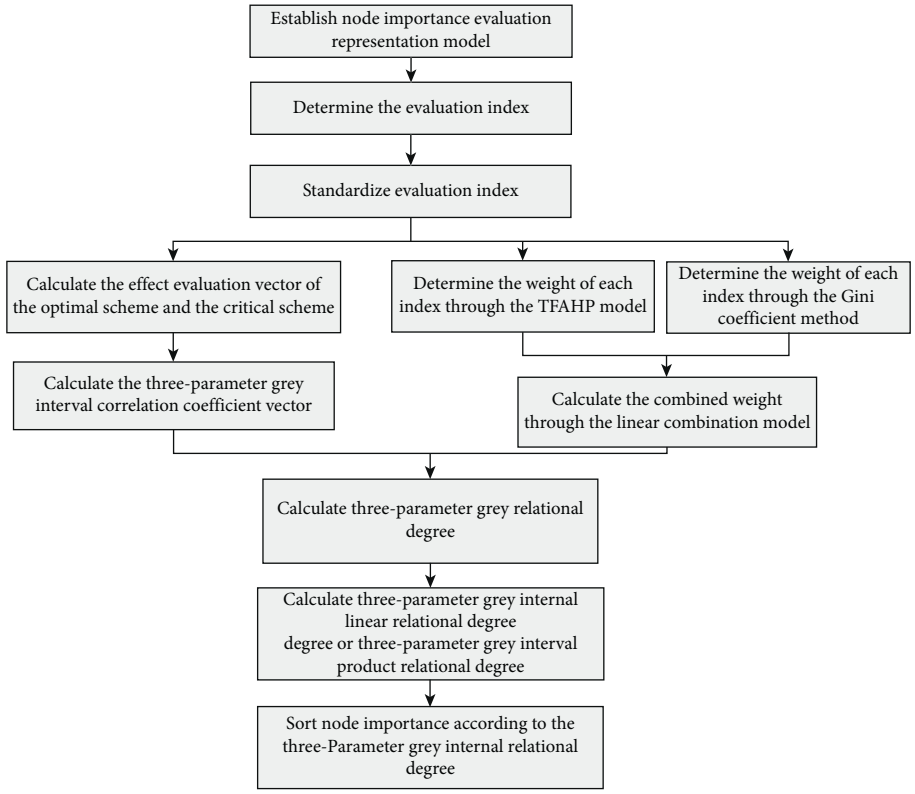


Figure 3: Three-parameter interval grey number grey relational evaluation model algorithm.

Figure 4: Large permanent magnet synchronous centrifugal unit.

5.3.4. Steps of Complex Product Node Importance Evaluation Based on Three-Parameter Grey Relational Degree. To sum up, the three-parameter interval grey relational evaluation algorithm of complex product node importance is as follows:

Step 1: construct the model of complex product presentation based on the multilayer complex network model.

Step 2: determine the evaluation index.

Step 3: standardize the original three-parameter interval grey number effect evaluation index, and obtain the standardized effect evaluation vector formula of each scheme.

Step 4: solve the ideal optimal solution for the decision-making problem and the effect evaluation vector $x^{+}(\otimes)$ and the critical solution $x^{-}(\otimes)$.

Step 5: obtain the three-parameter grey interval relational coefficient vector of each scheme, the ideal scheme and the critical scheme.

Table 3: Main parts and node name of permanent magnet synchronous centrifugal compressor.

Node	Parts
V_1	Principal axis
V_2	Inspiratory chamber
V_3	Import regulation structure
V_4	Impeller
V_5	Diffuser
V_6	Volute
V_7	Retaining ring
V_8	Tight ring
V_9	Thrust disk
V_{10}	Separator
V_{11}	Balance plate
V_{12}	Refluxer
V_{13}	Sealing
V_{14}	Bend
V_{15}	Bearing
V_{16}	Electrical machine
V_{17}	Lubricated gears
V_{18}	Bearing with lubrication
V_{19}	Curved casing
V_{20}	Pressure-regulating valve
V_{21}	Temperature sensor
V_{22}	Extraction cooler
V_{23}	Pressure sensor
V_{24}	Evaporator
V_{25}	Return evaporator
V_{26}	Lubricating oil
V_{27}	Cryogen
V_{28}	Oil cooler
V_{29}	Electronic expansion valve
V_{30}	Drier filter
V_{31}	Condenser

Table 4: Function knowledge network connection information.

Function knowledge network								
Source	Target	Weight	Source	Target	Weight	Source	Target	Weight
1	4	0.52	9	21	0.27	18	23	0.39
1	7	0.59	9	22	0.50	18	24	0.65
1	8	0.56	10	11	0.93	18	25	0.70
1	9	0.28	10	12	0.59	18	29	0.52
1	11	0.56	10	13	0.52	19	20	0.12
1	15	0.46	10	14	0.48	19	21	0.48
1	17	0.56	10	15	0.58	19	22	0.56
1	18	0.69	11	13	0.55	19	23	0.23
2	3	0.45	11	15	0.50	19	25	0.67
2	5	0.44	11	16	0.62	19	28	0.56
2	6	0.49	11	17	0.40	20	23	0.47
2	12	0.44	11	20	0.58	20	29	0.39
3	4	0.25	11	21	0.55	20	31	0.58
3	5	0.25	12	13	0.64	21	22	0.43
3	9	0.42	12	14	0.56	21	23	0.58
3	12	0.69	12	15	0.35	21	24	0.60
3	14	0.54	12	16	0.60	21	27	0.22
4	5	0.66	13	14	0.51	21	28	0.68
4	6	0.59	13	15	0.52	21	29	0.36
4	11	0.27	13	16	0.61	21	30	0.54
4	12	0.44	13	21	0.41	21	31	0.41
4	14	0.36	13	23	0.44	22	24	0.45
4	21	0.19	14	15	0.49	22	25	0.20
4	23	0.57	14	16	0.32	22	27	0.35
5	6	0.31	14	19	0.29	22	28	0.52
5	7	0.48	14	25	0.68	22	29	0.43
5	8	0.50	15	16	0.55	22	30	0.38
5	10	0.39	15	20	0.37	22	31	0.38
5	23	0.58	15	23	0.31	23	27	0.42
6	7	0.67	16	19	0.46	23	28	0.23
6	8	0.69	16	20	0.37	24	25	0.21
6	10	0.60	16	21	0.25	24	27	0.30
6	12	0.63	16	23	0.37	24	28	0.60
6	14	0.58	16	29	0.55	24	29	0.40
7	8	0.59	17	18	0.59	24	30	0.51
7	9	0.54	17	19	0.29	24	31	0.57
7	10	0.68	17	20	0.61	25	27	0.35
7	11	0.63	17	21	0.20	25	28	0.49
7	13	0.72	17	23	0.69	25	30	0.51
7	14	0.51	17	24	0.68	25	31	0.40
8	11	0.22	17	25	0.40	26	29	0.25
8	12	0.52	17	28	0.19	27	28	0.17
8	13	0.81	17	31	0.56	27	30	0.40
9	11	0.27	18	19	0.47	27	31	0.63
9	12	0.41	18	20	0.39	28	31	0.42
9	14	0.51	18	21	0.61	29	30	0.35
						30	31	0.68

Step 6: solve the TFAHP and Gini coefficient models, combine the weights, and obtain the weight of each scheme under different attributes.

Step 7: calculate the three-parameter interval grey number relational degree $G(x^{+}(\otimes), x_i(\otimes))$ and $G(x^{-}(\otimes), x_i(\otimes))$ $(i = 1, 2, \ldots, n)$ between each scheme and ideal scheme and critical scheme, and calculate the three-parameter grey interval linear relational degree or three-parameter grey interval product relational degree $G(x_i(\otimes))$ $(i = 1, 2, \ldots, n)$ of each scheme.

Step 8: the schemes are sorted according to the relevance degree $G(x_i(\otimes))$ $(i = 1, 2, \ldots, n)$. The scheme corresponding to the maximum relational degree is the optimal one.

The flow of this algorithm is shown in Figure 3.

6. Case Study

The high-speed permanent magnet synchronous frequency conversion centrifugal high-power chiller of G

Table 5: Behavior knowledge network connection information.

Behavior knowledge network								
Source	Target	Weight	Source	Target	Weight	Source	Target	Weight
1	6	0.45	9	11	0.34	17	21	0.63
1	7	0.34	9	12	0.27	17	23	0.50
1	8	0.43	9	13	0.37	17	24	0.53
1	9	0.56	9	21	0.44	17	25	0.79
1	11	0.28	9	23	0.60	18	19	0.29
1	15	0.33	10	11	0.64	18	20	0.60
2	3	0.53	10	13	0.56	18	21	0.26
2	4	0.50	10	14	0.22	18	25	0.45
2	6	0.16	10	15	0.32	19	20	0.79
2	13	0.43	11	13	0.49	19	21	0.46
3	4	0.37	11	15	0.46	19	22	0.23
3	5	0.53	11	16	0.32	19	25	0.66
3	9	0.62	11	17	0.31	20	23	0.49
3	12	0.50	11	20	0.36	20	29	0.28
3	14	0.52	11	21	0.66	21	22	0.32
4	5	0.22	12	13	0.50	21	23	0.59
4	6	0.52	12	14	0.42	21	24	0.50
4	11	0.47	12	15	0.51	21	30	0.45
4	12	0.67	12	16	0.29	21	31	0.26
4	13	0.59	13	14	0.26	22	24	0.46
4	20	0.61	13	15	0.69	22	25	0.27
4	21	0.64	13	16	0.53	22	29	0.23
5	6	0.63	13	21	0.36	22	30	0.68
5	7	0.59	13	23	0.53	22	31	0.24
5	8	0.56	14	15	0.21	23	27	0.33
5	9	0.26	14	16	0.30	23	28	0.41
5	21	0.21	14	19	0.53	24	25	0.34
6	7	0.61	14	25	0.58	24	27	0.42
6	9	0.51	15	16	0.63	24	29	0.76
6	10	0.67	15	20	0.61	24	30	0.41
6	12	0.69	15	23	0.70	24	31	0.31
6	13	0.46	16	19	0.67	25	27	0.51
7	8	0.33	16	20	0.26	25	28	0.32
7	9	0.44	16	21	0.23	25	31	0.36
7	11	0.23	16	22	0.61	26	29	0.56
7	12	0.40	16	28	0.21	27	30	0.42
7	13	0.39	17	18	0.66	27	31	0.50
7	14	0.61	16	28	0.21	28	31	0.57
8	10	0.34	17	18	0.66	29	30	0.46
8	12	0.35	17	19	0.49	30	31	0.37
8	13	0.41	17	20	0.31			

enterprise has been unanimously recognized as the world's first high-speed and high-power word synchronous frequency conversion centrifugal chiller after being certified by many experts. Its technology has reached the international leading level. Its design process will involve many parts, among which the parts have complex connections in many aspects. In this paper, multilayer network is used to sort out its relationship in detail, and the identification of key design parts is studied. The product drawing and component composition are shown in Figure 4 and Table 3.

First of all, combined with enterprise product knowledge base, design base, case base, and interviews with designers, we analyze the relationship between functional behavior and structure network of the large permanent magnet synchronous centrifugal unit. The multilayer network knowledge association is shown from Tables 4 to 6.

Through the calculation of index system, we can get the three-parameter interval grey number of the evaluation index as follows. This paper combines the enterprise product knowledge base, design library, and case library, as well as interviews with designers, to analyze the relationship and the

Table 6: Structure knowledge network connection information.

Structure knowledge network								
Source	Target	Weight	Source	Target	Weight	Source	Target	Weight
1	4	0.23	10	12	0.49	18	29	0.21
1	7	0.31	10	13	0.50	19	20	0.30
1	8	0.29	10	15	0.33	19	21	0.79
1	9	0.50	11	13	0.32	19	22	0.33
1	11	0.46	11	15	0.52	19	25	0.66
1	18	0.52	11	16	0.30	19	28	0.42
2	3	0.44	11	17	0.47	20	23	0.38
2	5	0.57	11	20	0.65	20	31	0.20
3	4	0.46	12	13	0.40	21	23	0.44
3	5	0.63	12	14	0.54	21	24	0.59
3	9	0.44	12	15	0.68	21	27	0.14
3	14	0.21	13	14	0.63	21	28	0.11
4	5	0.59	13	15	0.66	21	30	0.23
4	6	0.45	13	16	0.28	21	31	0.53
4	11	0.41	13	23	0.29	22	24	0.44
4	12	0.40	14	15	0.79	22	25	0.46
4	23	0.48	14	16	0.48	22	27	0.41
5	6	0.31	14	19	0.58	22	28	0.65
5	7	0.35	14	25	0.34	22	30	0.71
5	9	0.30	15	16	0.31	22	31	0.49
5	23	0.28	15	20	0.46	23	27	0.42
6	7	0.38	15	23	0.24	23	28	0.45
6	8	0.65	16	19	0.70	24	25	0.75
6	12	0.28	16	20	0.67	24	28	0.87
6	14	0.22	16	21	0.23	24	29	0.49
7	8	0.47	16	23	0.54	24	30	0.35
7	9	0.61	16	29	0.43	24	31	0.49
7	10	0.61	17	18	0.41	25	27	0.47
7	11	0.38	17	19	0.44	25	30	0.33
7	14	0.43	17	20	0.67	25	31	0.55
8	11	0.47	17	21	0.54	26	29	0.56
8	12	0.38	17	25	0.28	27	28	0.19
8	13	0.46	17	28	0.62	27	30	0.45
9	11	0.36	17	31	0.26	27	31	0.87
9	12	0.46	18	19	0.60	28	31	0.68
9	14	0.44	18	20	0.55	29	30	0.56
9	22	0.52	18	23	0.31	30	31	0.39
10	11	0.55	18	24	0.45			

FBS complex network between the parts of the large permanent magnet synchronous centrifugal unit. Then, the network characteristic analysis is carried out. According to the internal expert experience of the enterprise, the FBS multilayer network is assigned (1/3, 1/3, 1/3), (1/2, 1/4, 1/4), and (1/2, 1/4), 1/4) for obtaining three-parameter index data. The three-parameter value of the value information of the node is obtained through enterprise research. Then, we can get the three-parameter interval grey number of the evaluation index as follows:

$$X(\otimes)_O = \begin{bmatrix}
[5.2772\ \ 5.3755\ \ 5.4411] & [139\ \ 142.6\ \ 145] & [1.5701\ \ 1.7090\ \ 1.8016] & [2.2812\ \ 2.4658\ \ 2.5889] \\
[3.6959\ \ 4.0257\ \ 4.2455] & [43\ \ 47.2\ \ 50] & [2.9399\ \ 3.2129\ \ 3.3948] & [0.7282\ \ 0.7512\ \ 0.7665] \\
[3.8358\ \ 4.4389\ \ 4.8409] & [87\ \ 90\ \ 92] & [3.5253\ \ 3.7895\ \ 3.9656] & [0.8195\ \ 0.8455\ \ 0.8628] \\
[4.2019\ \ 4.3726\ \ 4.4863] & [128\ \ 131.6\ \ 134] & [3.4982\ \ 3.7958\ \ 3.9942] & [2.0347\ \ 2.1363\ \ 2.2040] \\
[2.0656\ \ 2.5827\ \ 2.9274] & [31\ \ 33.4\ \ 35] & [3.3349\ \ 3.6814\ \ 3.9124] & [0.3746\ \ 0.3837\ \ 0.3898] \\
[3.5186\ \ 3.4648\ \ 3.4290] & [32\ \ 36.2\ \ 39] & [3.1801\ \ 3.3571\ \ 3.4751] & [0.3470\ \ 0.3683\ \ 0.3825] \\
[3.2530\ \ 3.6687\ \ 3.9458] & [30\ \ 34.2\ \ 37] & [2.7420\ \ 3.0989\ \ 3.3368] & [0.9096\ \ 0.9405\ \ 0.9611] \\
[3.6148\ \ 3.7531\ \ 3.8453] & [21\ \ 25.2\ \ 28] & [2.0211\ \ 2.4675\ \ 2.7651] & [0.1458\ \ 0.4538\ \ 0.6591] \\
[3.5515\ \ 3.7544\ \ 3.8896] & [26\ \ 29.6\ \ 32] & [3.6857\ \ 3.8382\ \ 3.9398] & [0.9985\ \ 0.8067\ \ 0.6788] \\
[1.2876\ \ 1.8667\ \ 2.2528] & [23\ \ 27.8\ \ 31] & [3.8336\ \ 4.1062\ \ 4.2879] & [0.5379\ \ 0.5567\ \ 0.5692] \\
[2.3693\ \ 2.7321\ \ 2.9740] & [36\ \ 39\ \ 41] & [3.3977\ \ 3.8290\ \ 4.1166] & [0.8360\ \ 0.8581\ \ 0.8728] \\
[2.8436\ \ 3.2467\ \ 3.5154] & [38\ \ 41\ \ 43] & [2.7009\ \ 3.1348\ \ 3.4240] & [0.7279\ \ 0.7702\ \ 0.7985] \\
[2.1737\ \ 2.6398\ \ 2.9506] & [32\ \ 35\ \ 37] & [3.4680\ \ 3.6469\ \ 3.7662] & [0.9241\ \ 0.9612\ \ 0.9859] \\
[2.4560\ \ 2.7803\ \ 2.9964] & [26\ \ 29\ \ 31] & [3.6845\ \ 3.8422\ \ 3.9473] & [0.7017\ \ 0.7702\ \ 0.8158] \\
[1.9852\ \ 2.1486\ \ 2.2574] & [35\ \ 40.4\ \ 44] & [4.6799\ \ 4.8129\ \ 4.9015] & [0.8126\ \ 0.8492\ \ 0.8737] \\
[2.6777\ \ 3.2085\ \ 3.5624] & [29\ \ 32\ \ 34] & [3.1567\ \ 3.5532\ \ 3.8175] & [0.3707\ \ 0.3947\ \ 0.4107] \\
[1.4577\ \ 2.1785\ \ 2.6590] & [26\ \ 27.8\ \ 29] & [3.0778\ \ 3.4801\ \ 3.7483] & [0.6101\ \ 0.6628\ \ 0.6980] \\
[3.9226\ \ 4.2512\ \ 4.4703] & [76\ \ 83.8\ \ 89] & [2.9147\ \ 3.2387\ \ 3.4546] & [1.0031\ \ 1.1734\ \ 1.2869] \\
[4.6784\ \ 4.9707\ \ 5.1655] & [78\ \ 86.4\ \ 92] & [3.6177\ \ 4.0368\ \ 4.3161] & [1.0211\ \ 1.1884\ \ 1.2999] \\
[2.0672\ \ 2.5348\ \ 2.8466] & [30\ \ 33.6\ \ 36] & [3.9365\ \ 4.2157\ \ 4.4019] & [0.5793\ \ 0.5898\ \ 0.5968] \\
[2.5180\ \ 2.7985\ \ 2.9856] & [35\ \ 38.6\ \ 41] & [4.0827\ \ 4.5355\ \ 4.8373] & [0.4204\ \ 0.4333\ \ 0.4420] \\
[2.9706\ \ 3.4396\ \ 3.7522] & [37\ \ 40.6\ \ 43] & [3.4184\ \ 3.6110\ \ 3.7395] & [0.4923\ \ 0.5040\ \ 0.5118] \\
[2.6569\ \ 2.8298\ \ 2.9451] & [36\ \ 39.6\ \ 42] & [3.8562\ \ 4.0728\ \ 4.2173] & [0.4735\ \ 0.5034\ \ 0.5234] \\
[3.9019\ \ 4.4699\ \ 4.8486] & [55\ \ 58.6\ \ 61] & [2.9948\ \ 3.4043\ \ 3.6773] & [0.8101\ \ 0.8383\ \ 0.8572] \\
[3.1264\ \ 3.5869\ \ 3.8939] & [34\ \ 38.2\ \ 41] & [3.3830\ \ 3.7307\ \ 3.9625] & [0.7072\ \ 0.7874\ \ 0.8409] \\
[3.6097\ \ 4.1982\ \ 4.5906] & [30\ \ 34.2\ \ 37] & [3.3496\ \ 3.5309\ \ 3.6517] & [0.4995\ \ 0.5218\ \ 0.5367] \\
[3.8147\ \ 4.0691\ \ 4.2386] & [29\ \ 32\ \ 34] & [3.3914\ \ 4.1287\ \ 4.6202] & [0.5435\ \ 0.5653\ \ 0.5799] \\
[2.8457\ \ 3.2798\ \ 3.5692] & [28\ \ 30.4\ \ 32] & [3.3991\ \ 3.5278\ \ 3.6136] & [0.4307\ \ 0.4819\ \ 0.5161] \\
[3.1386\ \ 3.8243\ \ 4.2814] & [37\ \ 39.4\ \ 41] & [3.9917\ \ 4.1519\ \ 4.2587] & [0.5029\ \ 0.5433\ \ 0.5703] \\
[4.1264\ \ 4.7321\ \ 5.1358] & [29\ \ 32.6\ \ 35] & [3.6002\ \ 4.3466\ \ 4.8442] & [0.4190\ \ 0.4262\ \ 0.4310] \\
[3.6097\ \ 3.9617\ \ 4.1964] & [26\ \ 28.4\ \ 30] & [3.7239\ \ 4.1649\ \ 4.4590] & [0.5303\ \ 0.5474\ \ 0.5587]
\end{bmatrix}. \tag{19}$$

The normalized three-parameter interval grey number evaluation matrix is

$$X(\otimes =)\begin{bmatrix}
[0.9606\ 0.9842\ 1.0000] & [0.9516\ 0.9806\ 1.0000] & [1.0000\ 0.9634\ 0.9354] & [0.5516\ 0.6413\ 0.7011] \\
[0.5798\ 0.6592\ 0.7122] & [0.1774\ 0.2113\ 0.2339] & [0.5919\ 0.5096\ 0.4546] & [0.2830\ 0.2941\ 0.3016] \\
[0.6135\ 0.7587\ 0.8555] & [0.5323\ 0.5565\ 0.5726] & [0.4153\ 0.3356\ 0.2824] & [0.3273\ 0.3399\ 0.3484] \\
[0.7017\ 0.7427\ 0.7701] & [0.8629\ 0.8919\ 0.9113] & [0.4235\ 0.3337\ 0.2738] & [0.9178\ 0.9671\ 1.0000] \\
[0.1873\ 0.3118\ 0.3948] & [0.0806\ 0.1000\ 0.1129] & [0.4727\ 0.3682\ 0.2985] & [0.1112\ 0.1156\ 0.1185] \\
[0.5371\ 0.5242\ 0.5156] & [0.0887\ 0.1226\ 0.1452] & [0.5195\ 0.4660\ 0.4304] & [0.0978\ 0.1081\ 0.1150] \\
[0.4732\ 0.5733\ 0.6400] & [0.0726\ 0.1065\ 0.1290] & [0.6516\ 0.5440\ 0.4722] & [0.3711\ 0.3861\ 0.3961] \\
[0.5603\ 0.5936\ 0.6158] & [0.0000\ 0.0339\ 0.0565] & [0.8692\ 0.7345\ 0.6447] & [0.0000\ 0.1496\ 0.2494] \\
[0.5451\ 0.5939\ 0.6265] & [0.0403\ 0.0694\ 0.0887] & [0.3669\ 0.3209\ 0.2902] & [0.4143\ 0.3211\ 0.2590] \\
[0.0000\ 0.1394\ 0.2324] & [0.0161\ 0.0548\ 0.0806] & [0.3223\ 0.2400\ 0.1852] & [0.1905\ 0.1996\ 0.2057] \\
[0.2604\ 0.3478\ 0.4060] & [0.1210\ 0.1452\ 0.1613] & [0.4538\ 0.3236\ 0.2369] & [0.3354\ 0.3461\ 0.3532] \\
[0.3746\ 0.4717\ 0.5364] & [0.1371\ 0.1613\ 0.1774] & [0.6640\ 0.5331\ 0.4458] & [0.2828\ 0.3034\ 0.3171] \\
[0.2133\ 0.3256\ 0.4004] & [0.0887\ 0.1129\ 0.1290] & [0.4326\ 0.3786\ 0.3426] & [0.3781\ 0.3962\ 0.4082] \\
[0.2813\ 0.3594\ 0.4114] & [0.0403\ 0.0645\ 0.0806] & [0.3673\ 0.3197\ 0.2879] & [0.2701\ 0.3034\ 0.3255] \\
[0.1680\ 0.2073\ 0.2335] & [0.1129\ 0.1565\ 0.1855] & [0.0669\ 0.0268\ 0.0000] & [0.3240\ 0.3418\ 0.3536] \\
[0.3347\ 0.4625\ 0.5477] & [0.0645\ 0.0887\ 0.1048] & [0.5265\ 0.4069\ 0.3271] & [0.1093\ 0.1209\ 0.1287] \\
[0.0409\ 0.2145\ 0.3302] & [0.0403\ 0.0548\ 0.0645] & [0.5503\ 0.4289\ 0.3480] & [0.2256\ 0.2512\ 0.2683] \\
[0.6344\ 0.7135\ 0.7663] & [0.4435\ 0.5065\ 0.5484] & [0.5995\ 0.5018\ 0.4366] & [0.4165\ 0.4993\ 0.5544] \\
[0.8164\ 0.8868\ 0.9337] & [0.4597\ 0.5274\ 0.5726] & [0.3874\ 0.2609\ 0.1767] & [0.4253\ 0.5066\ 0.5607] \\
[0.1877\ 0.3003\ 0.3753] & [0.0726\ 0.1016\ 0.1210] & [0.2912\ 0.2069\ 0.1508] & [0.2106\ 0.2157\ 0.2191] \\
[0.2962\ 0.3638\ 0.4088] & [0.1129\ 0.1419\ 0.1613] & [0.2471\ 0.1105\ 0.0194] & [0.1334\ 0.1397\ 0.1439] \\
[0.4052\ 0.5181\ 0.5934] & [0.1290\ 0.1581\ 0.1774] & [0.4475\ 0.3894\ 0.3507] & [0.1683\ 0.1740\ 0.1778] \\
[0.3297\ 0.3713\ 0.3991] & [0.1210\ 0.1500\ 0.1694] & [0.3154\ 0.2501\ 0.2065] & [0.1592\ 0.1738\ 0.1834] \\
[0.6294\ 0.7662\ 0.8574] & [0.2742\ 0.3032\ 0.3226] & [0.5754\ 0.4518\ 0.3694] & [0.3227\ 0.3365\ 0.3456] \\
[0.4427\ 0.5536\ 0.6275] & [0.1048\ 0.1387\ 0.1613] & [0.4582\ 0.3533\ 0.2833] & [0.2728\ 0.3117\ 0.3377] \\
[0.5591\ 0.7008\ 0.7952] & [0.0726\ 0.1065\ 0.1290] & [0.4683\ 0.4136\ 0.3771] & [0.1719\ 0.1827\ 0.1899] \\
[0.6084\ 0.6697\ 0.7105] & [0.0645\ 0.0887\ 0.1048] & [0.4557\ 0.2332\ 0.0849] & [0.1932\ 0.2038\ 0.2109] \\
[0.3751\ 0.4796\ 0.5493] & [0.0565\ 0.0758\ 0.0887] & [0.4534\ 0.4145\ 0.3886] & [0.1384\ 0.1633\ 0.1799] \\
[0.4456\ 0.6107\ 0.7208] & [0.1290\ 0.1484\ 0.1613] & [0.2745\ 0.2262\ 0.1940] & [0.1735\ 0.1931\ 0.2062] \\
[0.6835\ 0.8293\ 0.9265] & [0.0645\ 0.0935\ 0.1129] & [0.3927\ 0.1674\ 0.0173] & [0.1327\ 0.1362\ 0.1385] \\
[0.5591\ 0.6438\ 0.7003] & [0.0403\ 0.0597\ 0.0726] & [0.3554\ 0.2223\ 0.1335] & [0.1868\ 0.1951\ 0.2006] \\
[0.9606\ 0.9842\ 1.0000] & [0.9516\ 0.9806\ 1.0000] & [1.0000\ 0.9634\ 0.9354] & [0.9178\ 0.9671\ 1.0000] \\
[0.0000\ 0.1394\ 0.2324] & [0.0000\ 0.0339\ 0.0565] & [0.0669\ 0.0268\ 0.0000] & [0.0000\ 0.1081\ 0.1150]
\end{bmatrix}. \quad (20)$$

Table 7: A-B comprehensive fuzzy judgment matrix.

A	$B1$			$B2$			$B3$			$B4$		
$B1$	0.5	0.5	0.5	0.292	0.342	0.392	0.517	0.567	0.617	0.463	0.513	0.563
$B2$	0.611	0.661	0.711	0.5	0.5	0.5	0.563	0.613	0.663	0.544	0.594	0.644
$B3$	0.392	0.442	0.492	0.341	0.391	0.441	0.5	0.5	0.5	0.433	0.483	0.533
$B4$	0.442	0.492	0.542	0.365	0.415	0.465	0.471	0.521	0.571	0.5	0.5	0.5

Table 8: Fuzzy complementary matrix.

	$B1$	$B2$	$B3$	$B4$
$B1$	0.500	0.342	0.567	0.513
$B2$	0.661	0.500	0.613	0.594
$B3$	0.442	0.391	0.500	0.483
$B4$	0.492	0.415	0.521	0.500

Table 9: Calculation results and comparison.

Node	$G(x^{+}(\otimes), x_i(\otimes))$	$G(x^{-}(\otimes), x_i(\otimes))$	Linear correlation degree	Ranking of this article
1	0.7227	0.5026	0.6101	2
2	0.6196	0.5950	0.5123	16
3	0.6542	0.6244	0.5149	15
4	0.8367	0.4228	0.7070	1
5	0.6910	0.4836	0.6037	3
6	0.5902	0.6887	0.4508	30
7	0.7086	0.6146	0.5470	11
8	0.5871	0.5846	0.5013	20
9	0.5933	0.5873	0.5030	19
10	0.6220	0.7173	0.4524	28
11	0.7488	0.7116	0.5186	13
12	0.6741	0.5362	0.5690	5
13	0.6233	0.7190	0.4522	29
14	0.8176	0.7023	0.5577	8
15	0.8065	0.6980	0.5543	9
16	0.7319	0.5711	0.5804	4
17	0.7181	0.7309	0.4936	25
18	0.6902	0.6997	0.4953	23
19	0.6756	0.6864	0.4946	24
20	0.7951	0.6931	0.5510	10
21	0.6626	0.7156	0.4735	26
22	0.6771	0.6590	0.5091	17
23	0.6260	0.6117	0.5072	18
24	0.6172	0.4806	0.5683	6
25	0.6611	0.6591	0.5010	21
26	0.6777	0.6429	0.5174	14
27	0.6695	0.7229	0.4733	27
28	0.6793	0.7852	0.4470	31
29	0.6231	0.6245	0.4993	22
30	0.6583	0.6132	0.5225	12
31	0.6755	0.5465	0.5645	7

Table 10: Ranking of node importance calculated in [17].

Node	Correlation degree calculated in [17]	Ranking of [17]
1	0.6842	2
2	0.5933	16
3	0.5984	14
4	0.6952	1
5	0.6807	3
6	0.5163	30
7	0.6125	10
8	0.5705	20
9	0.5859	18
10	0.5328	27
11	0.6020	13
12	0.6507	5
13	0.5246	29
14	0.6214	9
15	0.6267	7
16	0.6690	4
17	0.5376	26
18	0.5505	23
19	0.5680	21
20	0.6078	11
21	0.5263	28
22	0.5823	19
23	0.5931	17
24	0.6215	8
25	0.5570	22
26	0.5949	15
27	0.5430	24
28	0.5023	31
29	0.5420	25
30	0.6023	12
31	0.6377	6

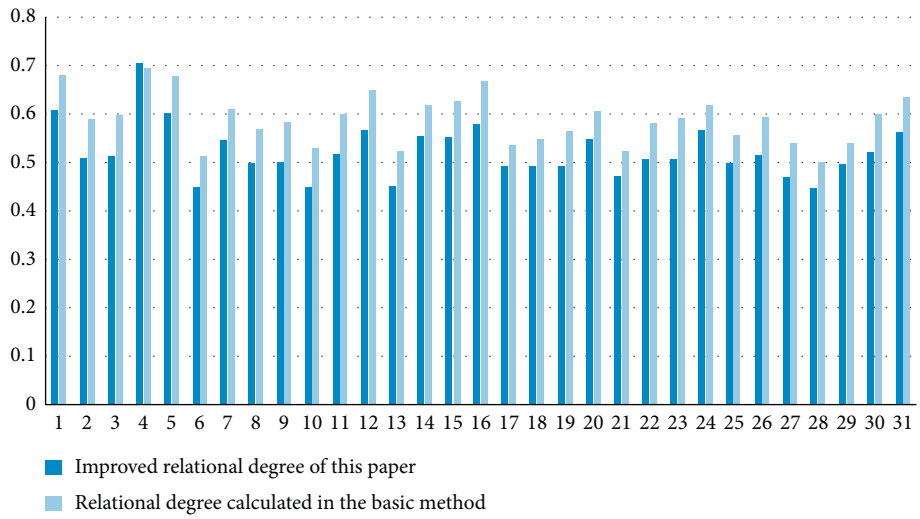


Figure 5: Comparison of relational degree results.

According to formula (4), the effect evaluation vectors of ideal optimal scheme and critical scheme are obtained:

$$\begin{aligned} X^{+}(\otimes) &= ([0.9606\ \ 0.9842\ \ 1.0000], [0.9516\ \ 0.9806\ \ 1.0000], [1.0000\ \ 0.9634\ \ 0.9354], [0.9178\ \ 0.9671\ \ 1.0000]), \\ X^{-}(\otimes) &= ([0.0000\ \ 0.1394\ \ 0.2324], [0.0000\ \ 0.0339\ \ 0.0565], [0.0669\ \ 0.0268\ \ 0.0000], [0.0000\ \ 0.1081\ \ 0.1150]). \end{aligned} \tag{21}$$

Then, we use TFAHP to calculate the weight:

The A-B comprehensive fuzzy judgment matrix is obtained, as shown in Table 7.

The fuzzy complementary matrix is shown in Table 8.

The relative weights of several indexes obtained by TFAHP are as follows:

$$W = (w_1, w_2, w_3, w_4) = (0.19, 0.32, 0.21, 0.28). \tag{22}$$

The weight obtained from the Gini coefficient is as follows: $W = (w_1, w_2, w_3, w_4) = (0.24, 0.29, 0.17, 0.30)$.

Then, we can calculate the comprehensive weight $W = (w_1, w_2, w_3, w_4) = (0.235, 0.30, 0.18, 0.285)$.

According to equations (8) and (10), the grey interval relational degree of each scheme with the ideal optimal scheme and the critical scheme can be obtained. Then, the linear relational degree is calculated by formula (5) (according to [28] and experts' opinion, $\lambda = 0.5$, $\xi = 0.5$, and $\beta_1 = \beta_2 = 0.5$).

And, we also calculate the node importance ranking through the method of [17]. The above results are shown in Tables 9 and 10 and Figure 5.

According to the method of [17], the ranking of node importance is shown in Table 10.

Then, we compare the results of the two methods, as shown in Figure 5.

From the calculation results of this method and the comparison with other methods, we can see that the calculation results of the two methods are in the top several, respectively, impeller ≻ principal axis ≻ diffuser ≻ electrical machine ≻ evaporator ≻ evaporator ≻ condenser ≻ bend and impeller ≻ principal axis ≻ diffuser ≻ electrical machine ≻ refluxer ≻ condenser ≻ bearing ≻ evaporator. We can see that the ranking results of the two are roughly the same, except for some subtle differences. The ranking results are realistic and roughly the same. However, we can see from the graphic difference in Figure 5 that the ranking difference obtained by the method proposed in this paper is stronger. This method makes the ranking more obvious and easy to identify. The evaluation method proposed in this paper increases the operability and identifiability of the importance ranking of design elements in the design process.

7. Conclusion

In order to comprehensively describe the complex product design process, this paper uses the complex product FBS model to describe the complex detailed design relationship. Then, the identification index system of important parts in the complex product design stage is constructed according to the characteristics of network. The three-parameter interval grey number relational model is improved based on the fuzzy analytic hierarchy process. The comparison of experimental results shows that the results calculated by the proposed method are more representative. It also shows that the method is practical and effective.

The multilevel description model of the design process based on FBS proposed in this paper can describe the design process in more detail, and the analysis of the detailed design process is more specific. The evaluation of important parts based on this is more scientific. In addition, the weight improvement of the three-parameter interval grey number based on the fuzzy analytic hierarchy process and the Gini coefficient method is in line with the law of actual data and is more effective. The research comparison of this paper also shows the effectiveness and feasibility of the proposed method.

The next research prospect of this paper is to further improve the objectivity of data acquisition in the design stage and reduce the proportion of experience data of designers. Subsequently, we will try to establish a more scientific design database, standardize the management of its design process, and make its data acquisition more accurate and scientific. It will provide service structure for each department to obtain information, which makes the research more practical.

Data Availability

All data generated and materials or analyzed during this study are included in this published article.

Ethical Approval

This article does not involve the content of violating ethics and morality.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Weiming Yang, Congdong Li, and Yinyun Yu conceptualized the study; Weiming Yang and Congdong Li proposed the model; Weiming Yang and Mingsheng Zhong carried out the case study; Weiming Yang, Congdong Li, Yinyun Yu, and Mingsheng Zhong wrote, reviewed, and edited the manuscript. All authors have read and agreed to participate and publish the manuscript.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 72072072), National Natural Science Foundation of China (no. 71672074), Natural Science Foundation of Guangdong Province of China (no. 2019A1515010045), and 2018 Guangzhou Leading Innovation Team Program, China (no. 201909010006).

References

[1] L. Luo, J. Wang, and K. Zhang, "A model for complex product development performance based on knowledge management," *Operations Research and Management Science*, vol. 26, no. 1, pp. 166–172, 2017.

[2] R. Magnaye, B. Sauser, P. Patanakul, D. Nowicki, and W. Randall, "Earned readiness management for scheduling, monitoring and evaluating the development of complex product systems," *International Journal of Project Management*, vol. 32, no. 7, pp. 1246–1259, 2014.

[3] C.-Y. Huang, C.-L. Lee, T.-H. Wen, and C.-T. Sun, "A computer virus spreading model based on resource limitations and interaction costs," *Journal of Systems and Software*, vol. 86, no. 3, pp. 801–808, 2013.

[4] M. Jalili and M. Perc, "Information cascades in complex networks," *Journal of Complex Networks*, vol. 5, no. 5, pp. 665–693, 2017.

[5] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: a survey of spreading processes on complex networks," *IEEE Control Systems*, vol. 36, no. 1, pp. 26–46, 2016.

[6] F. Lu, W. Zhang, L. Shao, X. Jiang, P. Xu, and H. Jin, "Scalable influence maximization under independent cascade model," *Journal of Network and Computer Applications*, vol. 86, pp. 15–23, 2017.

[7] M. Gong, J. Yan, B. Shen, L. Ma, and Q. Cai, "Influence maximization in social networks based on discrete particle swarm optimization," *Information Sciences*, vol. 367-368, pp. 600–614, 2016.

[8] Z.-K. Bao, J.-G. Liu, and H.-F. Zhang, "Identifying multiple influential spreaders by a heuristic clustering algorithm," *Physics Letters A*, vol. 381, no. 11, pp. 976–983, 2017.

[9] L. Z. Cui, H. X. Hu, S. Yu et al., "DDSE: a novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks," *Journal of Network and Computer Applications*, vol. 103, pp. 976–983, 2018.

[10] H. Li, R. Zhang, Z. Zhao, and X. Liu, "LPA-MNI: an improved label propagation algorithm based on modularity and node importance for community detection," *Entropy*, vol. 23, no. 5, p. 497, 2021.

[11] Z. N. Berberler, H. I. Yildirim, T. Iltuzer, and I. Tunc, "Agglomeration-based node importance analysis in wheel-type networks," *International Journal of Foundations of Computer Science*, vol. 32, no. 3, pp. 269–288, 2021.

[12] Q. Sun, G. Yang, and A. Zhou, "An entropy-based self-adaptive node importance evaluation method for complex networks," *Complexity*, vol. 2020, Article ID 4529429, 13 pages, 2020.

[13] F. Bencherif and L. H. Mouss, "Complex network to enhance characterization analysis in modelling product development process," *African Journal of Science, Technology, Innovation and Development*, vol. 12, no. 7, pp. 797–811, 2020.

[14] R. Li, N. Yang, Y. Zhang, H. Liu, and M. Zhang, "Impacts of module-module aligned patterns on risk cascading propagation in complex product development (CPD) interdependent networks," *Physica A-Statistical Mechanics and Its Applications*, vol. 564, 2021.

[15] R. Li, H. Yi, and H. Cao, "Towards understanding dynamic design change propagation in complex product development via complex network approach," *International Journal of Production Research*, vol. 18, 2021.

[16] T. Li, H. Chen, J. Yuan, J. Qian, and A. W. Siyal, "Quality risk propagation of complex product collaborative manufacturing supply chain network based on CN and SoV," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 8889903, 16 pages, 2020.

[17] X. Yin, Y. Mo, C. Dong, and Y. Zhang, "Identification of the influential parts in a complex mechanical product from a reliability perspective using complex network theory," *Quality and Reliability Engineering International*, vol. 36, no. 2, pp. 604–622, 2020.

[18] G. Yu, Y. Yang, X. Zhang, and C. Li, "Network-based analysis of requirement change in customized complex product development," *International Journal of Information Technology and Decision Making*, vol. 36, pp. 604–622, 2020.

[19] D. Zhang, Z. Zhao, Y. Zhou, and Y. Guo, "A novel complex network-based modeling method for heterogeneous product design," *Cluster Computing-the Journal of Networks Software Tools and Applications*, vol. 22, pp. S7861–S7872, 2019.

[20] N. Zhang, M. Li, H. Ren, and Y. Li, "A network-based four-phase routing approach of multisource design change propagation on complex products," *Kybernetes*, 2020, in press.

[21] N. Zhang, Y. Yang, Y. Zheng, and J. Su, "Module partition of complex mechanical products based on weighted complex networks," *Journal of Intelligent Manufacturing*, vol. 30, no. 4, pp. 1973–1998, 2019.

[22] Y. Li and D. Zhang, "Multi-attribute group grey target decision-making method based on three-parameter interval grey number," *Journal of Grey System*, vol. 32, pp. 96–109, 2020.

[23] M. Li, K. Chen, C. Liu, and B. Liu, "The interval parameter optimization model based on three-way decision space and its application on "green products recommendation," *Discrete Dynamics in Nature and Society*, vol. 2020, 2020.

[24] K. Chen and P. Chen, "Decision making method of TOPSIS based on three-parameter interval grey numbers," *Systems Engineering and Electronics*, vol. 41, pp. 124–130, 2019.

[25] Y. Li and D. Zhang, "Dynamic multi-attribute decision-making method with three-parameter interval grey number based on the prospect theory," *Grey Systems: Theory and Application*, vol. 8, no. 4, pp. 424–435, 2018.

[26] Y. Li, Y. Niu, W. Wang, and B. Li, "Grey-incidence Clustering Decision-making method with Three-parameter interval grey number based on Regret theory," in *Proceedimgs of the 2017 International Conference on Grey Systems and Intelligent Services (GSIS)*, IEEE, Stockholm, Sweden, August 2017.

[27] C. Li and J. Yuan, "A new multi-attribute decision-making method with three-parameter interval grey linguistic variable," *International Journal of Fuzzy Systems*, vol. 19, no. 2, pp. 292–300, 2017.

[28] X. Wang and Y. G. Dang, "Dynamic multiattribute decision-making methods with three-parameter interval grey number," *Control and Decision*, vol. 30, no. 09, pp. 1623–1629, 2015.

[29] D. Luo, "Decision-making method with three-parameter interval grey number," *System Engineering-Theory & Practice*, vol. 29, no. 01, pp. 124–130, 2009.

[30] S. L. Yan, S. F. Liu, J. J. Zhu, Z. G. Fang, and J. Liu, "TOPSIS decision-making method with three-parameter interval number based on entropy measure," *Chinese Journal of Management Science*, vol. 21, pp. 145–151, 2013.

[31] J. Y. Dong, S. P. Wan, and S. M. Chen, "Fuzzy best-worst method based on triangular fuzzy numbers for multi-criteria decision-making," *Information Sciences*, vol. 547, no. 08, pp. 1080–1104, 2021.

[32] S. P. Wan, Z. H. Chen, and J. Y. Dong, "An integrated interval type-2 fuzzy technique for democratic-autocratic multi-criteria decision making," *Knowledge-Based Systems*, vol. 214, Article ID 106735, 2021.

[33] S. P. Wan and J. Y. Dong, "A novel extension of best-worst method with intuitionistic fuzzy reference comparisons," *IEEE Transactions on Fuzzy Systems*, p. 1, 2021, https://ieeexplore.ieee.org/document/9373994.

[34] S. P. Wan, J. Y. Dong, and S. M. Chen, "Fuzzy best-worst method based on generalized interval-valued trapezoidal fuzzy numbers for multi-criteria decision-making," *Information Sciences*, vol. 573, pp. 493–518, 2021.

[35] M. H. Lu and K. P. Zhu, "Improved analytic hierarchy process based on triangular fuzzy number," in *Proceedings of the 2nd International Conference on Education, Economics and Management Research (ICEEMR)*, Singapore, June 2018.

[36] L. Jiang, T. Tao, C. Zhang, H. Jiang, and J. Wang, "Summary of the port shoreline resource evaluation based on triangular fuzzy analytic hierarchy process," *Polish Maritime Research*, vol. 24, no. s3, pp. 16–22, 2017.

[37] J. Correa-Parra, J. Francisco Vergara-Perucich, and C. Aguirre-Nunez, "Water privatization and inequality: Gini coefficient for water resources in Chile," *Water*, vol. 12, no. 12, Article ID 3369, 2020.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 2296605, 11 pages
https://doi.org/10.1155/2021/2296605

Research Article

A New Early Rumor Detection Model Based on BiGRU Neural Network

Xiangning Chen,[1,2] Caiyun Wang,[1,2] Dong Li,[1,2] and Xuemei Sun[1,2]

[1]*School of Computer Science and Technology, Tiangong University, Tianjin 300387, China*
[2]*Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems, Tianjin, China*

Correspondence should be addressed to Xuemei Sun; sunxuemei@tiangong.edu.cn

Received 20 May 2021; Accepted 4 August 2021; Published 18 August 2021

Academic Editor: Juan L. G. Guirao

With the progress of society and the rapid development of computer technology, rumors arise on social media, which seriously affects the social economy. How to detect rumors accurately and rapidly has become one hot research topic. In this paper, a new early rumor detection model is proposed. The aim of this model is to increase the efficiency and the accuracy of rumor detection simultaneously. Specifically, in this model, the input data is firstly refined through account filtering and data standardization, then the BiGRU is used to consider the context relationship, and a reinforcement learning algorithm is applied to detection. Experimental results show that compared with other early rumor detection models (e.g., checkpoints), the accuracy of the proposed model is improved by 0.5% with the same speed, which testifies the effectiveness of this model.

1. Introduction

Rumors refer to statements that have no corresponding factual basis but are fabricated and promoted through certain means. In today's highly developed situation of information dissemination media, it can spread quickly through social media, and malicious rumors may affect economy and society significantly. The negative impact of rumors may increase significantly when certain major events occur, such as the traceability of COVID-19 in 2019. This makes people realize that if malicious rumors are not discovered in time, they may continue to cause significant damage, so the timing of their detection is crucial. Figure 1 shows an example of a rumor propagating on Twitter which is named "German Wings Crash." The source message started a claim about the crash could be an Airbus A320 German wings. A German wing Airbus A320 with 150 people on board crashed in Barcelonite, southern France, with no one surviving. The message was retweeted by multiple users on Twitter, either by reposting, commenting, or questioning the original source message. We extracted several related tweets within 24 hours.

Currently, the most research on rumors uses the Twitter social platform and Weibo platform as the main research objects. Considering that the research of rumor detection technology on the Weibo dataset has been relatively mature, and the latest accuracy rate has reached about 95%, this paper uses public standard Twitter dataset as the main research object. Unlike funny videos and celebrity gossips that are popular on Weibo, hot social events are the most popular topics on Twitter.

Based on the platform of Twitter, many researchers have conducted research on rumor detection. When most researchers focus on improving the accuracy of rumor detection, Ma et al. [1] and Kwon et al. [2] have proposed the use of presettings in recent years. The method of fixed checkpoints can evaluate the timeliness of the discovery of rumors, but this method has the disadvantage of not being able to capture the changes in different rumors spreading modes. On this basis, Farajtabar et al. [3] proposed a combination of reinforcement learning and a point process network activity model to detect false news and achieved good results. This assessment of the timeliness of rumor detection is also one of the focuses of this paper.

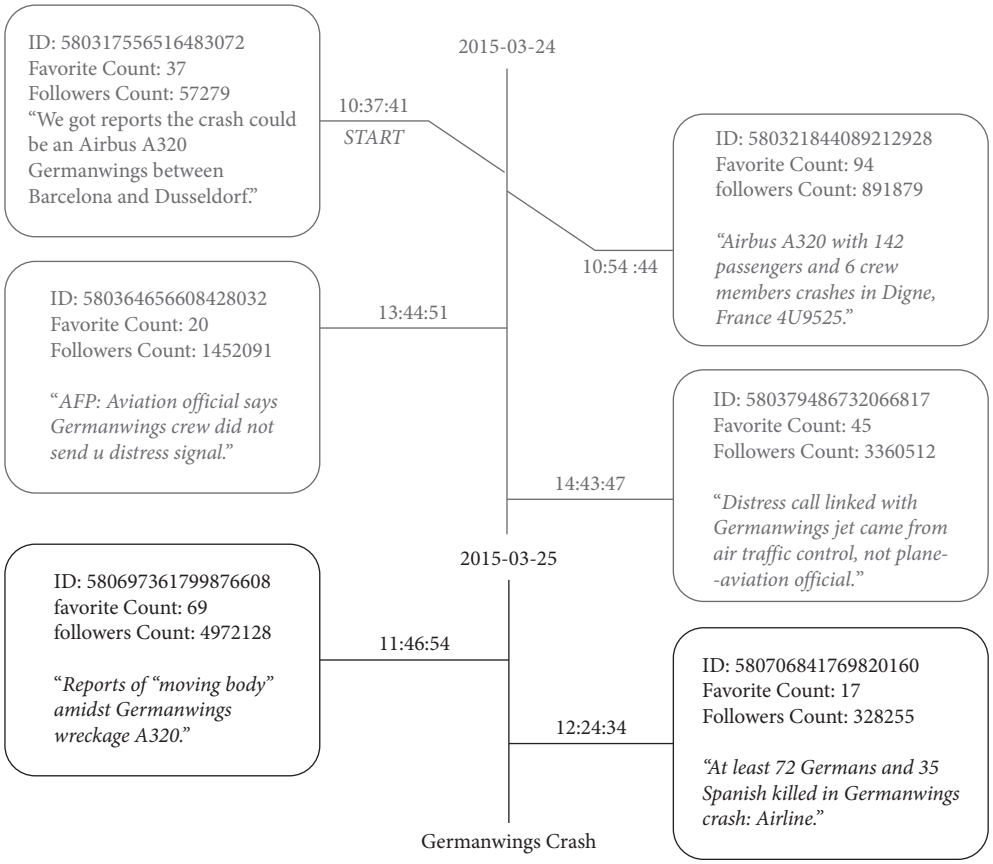


Figure 1: Rumors spread about the schematic "German Wings Crash" events (excerpt).

At the same time, although data preprocessing has been widely used in NLP (Natural Language Processing) to improve performance, few researchers conduct preprocessing on target data for rumor detection. Moreover, this paper finds that most studies focus on the extraction of various features the Twitter standard dataset to capture rumor indicators but ignore the data complexity on Twitter as a large social platform. Firstly, Twitter has a large number of spam accounts. The tweets sent by these accounts usually have a certain commercial purpose, and the research on rumor detection for these tweets is usually weakly relevant or invalid. They are intended to be included in the tweets for this paper. The extracted value information has caused serious interference, so it is necessary to filter it out. Secondly, the short text and randomness of tweets in Twitter makes it difficult for people and computers to accurately understand the information of tweets, so it is necessary to preprocess the tweets to standardize the tweets on the Twitter standard dataset.

The key contributions of our work are summarized as follows:

(1) In rumor detection, for the early rumor detection with checkpoints, this paper proposes to apply BiGRU to the early rumor detection model. Using BiGRU to consider the characteristics of context relations, combined with the sequence before and after, the two posts input before and after the event are included in the detection, so as to improve the effect of rumor detection.

(2) In rumor detection, aiming at the early rumor detection with checkpoints, this paper proposes a data preprocessing method based on account filtering and text standardization for the first time. The account filtering method is used to remove the junk account. Text standardization is used to standardize tweets in Twitter standard dataset so that the data can better express the meaning of the text so as to improve the accuracy of rumor detection.

(3) In this paper, Q-learning, a reinforcement learning algorithm, is applied for rumor detection to dynamically determine checkpoints, thereby improving the timeliness of rumor detection.

2. Related Work

In the contemporary era of the emergence of various social media, rumor detection has attracted the attention of all parties, and the current research on rumor detection has achieved initial results. The current research on rumor detection is mainly divided into two categories: one is based on traditional machine learning for rumor detection, and the other is based on feature learning to extract main features for rumor detection.

Rumor detection methods based on traditional machine learning mainly use decision trees, SVM, and other classifiers to classify events. Liang et al. [4] no longer use a single classifier, such as KNN and SVM but propose a BP neural model and an improved excitation function and add an impulse term to make it possible to detect rumors in the propagation process. Lu et al. [5] noticed that there is an imbalance in the data, which has affected the implementation of rumor detection and proposed a Co-Forest algorithm to improve the imbalanced data and balance the data distribution. Mao et al. [6] used an integrated classifier to detect rumors based on characteristics such as emotional orientation and communication process. In recent years, the rise of artificial intelligence has made the application of deep learning increasingly widespread. Similarly, in the field of rumor detection, deep learning plays an important role. Takahashi and Igata [7] developed a system to detect rumors by studying the spread of rumors and conducted experiments on Twitter and found that they can effectively detect rumors, which opened up a new chapter in rumor detection. Karamchandani and Franceschetti [8] proposed a method of detecting the source of rumors to control the rumors, which extended the best estimator of rules and irregularities to achieve the purpose of detecting the source of rumors. Similarly, Wu et al. [9] also considered propagation in the rumor detection technology and proposed a hybrid SVM classifier based on the graph kernel, which not only captures semantic features such as topics and emotions but also captures high. The first-order propagation mode improves the classification accuracy.

With the rise of artificial intelligence, rumor detection has entered a new stage. The method of rumor detection based on feature learning mainly uses advanced artificial intelligence ideas. Li et al. [10] combine the convolutional layer of the convolutional neural network to extract text features, use the GRU network to process the features, and then judge whether it is a rumor. Ren et al. [11] considered that Weibo text is a graph structure, and information such as the attitude of users' comments will affect the spread of Weibo text and proposed a rumor detection model based on time series. Liao et al. [12] considered the potential information of some Weibo texts and partial user information and proposed a social media rumor detection method based on a hierarchical attention network. Srinivasan and Dhinesh Babu [13] proposed a double convolutional neural network method with a new activation function for the sparse data with little available information that can be used to distinguish rumors at the beginning, because this method has faster generalization speed and more high precision and has a very good effect in rumor detection. After summarizing many studies, Zhou et al. [14] found that their research seldom considered the timeliness of rumor detection and proposed an early rumor detection model to process rumors through two modules, a rumor detection module and a checkpoint module. The rumor detection module is used to extract features, and the checkpoint module is used to solve the problem of timeliness, which is used to trigger the rumor detection module to ensure the timeliness of rumor detection while ensuring accurate identification of rumors. Lin et al. [15] raised the issue of word independence and found that some common words appear in rumors. Once these words appear, they can be judged as rumors. They proposed a deep sequence model to consider the two aspects of rumors: falsehood sex and influence, using long- and short-term memory units to learn falsehood, and combining deep sequences and social characteristics to learn influence. Asghar et al. [16] proposed a bidirectional long-term short-term memory model based on convolutional neural network, which uses convolutional neural network to extract post features and uses bidirectional long-term short-term memory method to store points and consider contextual information, effectively detecting rumors on Weibo, focusing on the research of rumor detection in Arabic, extracting information from users and content, and proposing the use of semisupervised expectation maximization (EM) to train newsworthy tweets topics to achieve the purpose of rumor detection [17].

In recent years, the research on rumor detection has mainly focused on extracting features and analyzing features. According to the text content of the given data, the main features expressed by the data are extracted and analyzed. Since the data comes from Twitter accounts, there are some spam accounts that guide public opinion, which interferes with the process of rumor detection and affects the results of the detection. At the same time, the longer the rumors spread, the more harmful it will be to the society, and timely detection of rumors is an important aspect of rumors detection research. Aiming at two aspects, this paper starts from the two directions of preprocessing and timeliness. On the one hand, the accuracy of rumor detection is improved through account screening and tweet standardization. On the other hand, reinforcement learning algorithms are used to save the time of rumor detection as much as possible.

3. DDR Model Architecture

The model mainly consists of three submodels: a data preprocessing model based on account filtering and standardization (DP model), a rumor detection model based on deep learning (DL model), and a checkpoint model based on reinforcement learning (RL model). Therefore, the proposed model is called DDR (data preprocessing and rumor detection based on deep learning and reinforcement learning) model for short as shown in Figure 2. In the DP model, this paper proposes basic information of users to analyze users, filter spam accounts in Twitter data, refine the data, and use standardized Twitter text to enhance the data to achieve accurate detection of rumors purpose. In the DL model, data characteristics are mainly analyzed, and BiGRU is used to consider the context, analyzes Twitter text data, and detects rumors. The RL model mainly solves the problem of the timeliness of rumors detection. The Q-learning algorithm is used to judge the detection results, and the reward and punishment mechanism is set to make the model trade-off between timeliness and accuracy to improve the timeliness of detection. At the same time, the accuracy of detection is improved.

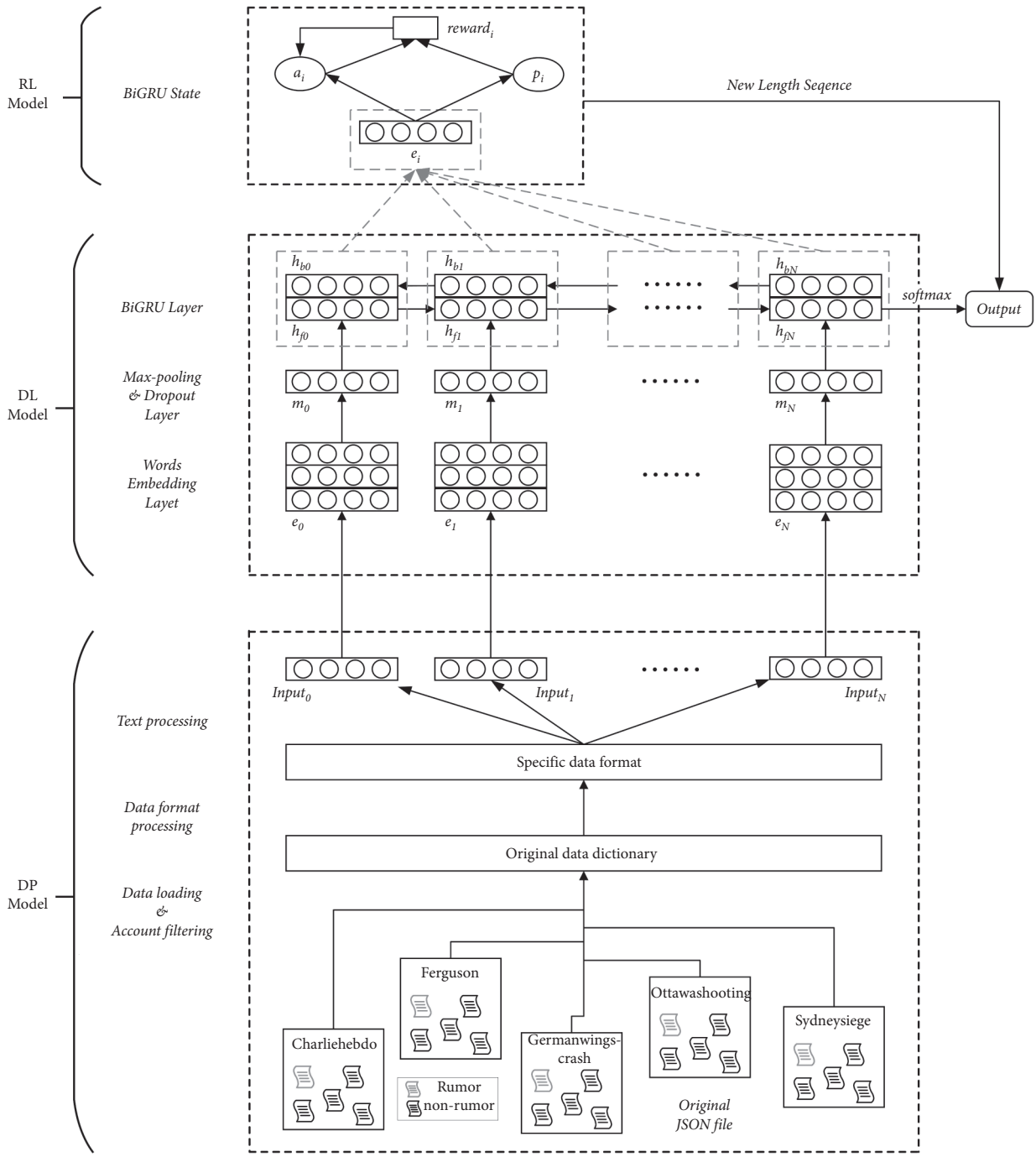


Figure 2: Architecture of DDR (data preprocessing and rumor detection based on deep learning and reinforcement learning).

3.1. DP Model

3.1.1. Account Filtering. For a Twitter account, its main features are generally divided into features based on user portraits and features based on tweet. User portrait features are a series of features that can be directly extracted from user information, such as the number of people that users follow and number of fans. User tweet features are statistical features extracted from user tweets, such as the proportion of tweets containing URLs and the proportion of liked tweets to the total tweets. Since the tweets of a single Twitter account in the dataset used for rumor detection usually is not larger enough to extract the user-based tweet text features, it is impossible to obtain complete user tweet features. At the same time, if a tweet of a single Twitter account has a high number of likes, the tweet can be considered valuable, thereby reducing the probability of the account being filtered.

Based on the above description, it can be concluded that when a certain type of Twitter account meets certain conditions, the tweets belonging to this account are usually weakly relevant or invalid for rumor detection. In this paper,

this type of account is defined as "filtered account," or FA, and using the user's portrait features and the number of likes on user tweets to determine whether an account is FA. The definition of FA is as follows:

(1) An FA will follow many people, while few users will follow the FA. Based on this feature, the authenticity definition of user accounts is proposed, as shown in the following:

$$\text{authenticity} = \frac{\text{FollowersCount}}{\text{FollowersCount} + \text{FriendsCount}}, \quad (1)$$

where FollowersCount denotes user followers and FriendsCount refers to the number of users following.

(2) Generally, the personal information of spam accounts in Twitter is not complete enough, and it is rare to fill in user description and user location information. Therefore, HasDesc and HasLoc are defined to indicate whether the user has description information and location information. Thus, HasDesc or HasLoc is 1; otherwise, it is 0. Based on the above analysis, the definition of user authority is proposed, as shown in the following:

$$\text{authority} = \text{authenticity} + 0.5 \cdot (\text{HasDesc} + \text{HasLoc}) + \frac{\text{TweetsLike}}{\text{AvgLike}}, \quad (2)$$

where TweetsLike denotes the total number of likes corresponding to the current user's tweet and AvgLike refers to the average of the total number of likes of all users.

(3) The authority of all users is sorted in a nonincreasing manner. It is concluded that the bottom 5% of users in the ranking have lower authority, and these users are defined as FA.

We filter out the FAs in the Twitter standard dataset and filter the tweets belonging to these FAs.

3.1.2. Tweet Standardization. The standardization of text is a part of text preprocessing, which mainly refers to the correction of some irregularities or errors in the text, thus transferring it to a text that people can understand correctly. Based on the characteristics of tweets in the Twitter standard dataset, a standardization method for tweet text is proposed in this section.

Tweets in Twitter are usually random and short. On the one hand, tweets are generally limited to 140 words. On the other hand, compared with traditional standard texts, tweets contain many irregularities or errors in terms of wording, grammar, format, and so on, such as spoken language, colloquialisms, acronyms, Internet terms, or emoji expressions, which greatly increase the difficulty of computer understanding of the text, disturbing factors in the difficulty of understanding the text. At the same time, the tweets also contain some symbols and network links that have no actual meaning, and other factors that have no relation with the semantics of the text.

In view of the above characteristics of tweets, in order to strengthen the computer's understanding of tweets, this paper carries out the following standardized processing on tweets:

(1) Unit replacement is as follows: replacing the unit in the text with a unified format, such as replacing "4 kgs" and "4 kg" with "4 kg"

(2) Acronym replacement is as follows: replacing the acronyms in the text with complete words, such as replacing "can't" with "can not"

(3) Spelling proofreading is as follows: replacing some network terms or punctuation of words with irregular spelling, such as replacing "rep" with "reply"

(4) Punctuation is as follows: adding spaces on both sides of all punctuation

(5) Symbol replacement is as follows: replacing all logical symbols with words, such as "and" with the word "and"

(6) Redundant information processing is as follows: removing extra spaces, "@" and "#" symbols in hashtags and removing all hyperlink information

(7) Delete stopwords is as follows: deleting a series of stop words such as "if" and "to"

(8) Part-of-speech restoration is as follows: restoring an English word of any form to its general form, such as "does," "did," and "done" unified reduction to "do"

3.2. DL Model. Rumor detection model based on deep learning processes the tweets after tweet standardization, dividing into words embedding layer, max-pooling and dropout layer, and BiGRU layer. It is used to transform a piece of text into the final state and to judge whether the text is a rumor through the softmax function.

3.2.1. Words Embedding Layer. In the words embedding layer, this paper first performs word segmentation on the text Input_i that has been standardized. Considering that simple splitting will destroy the semantics of compound words such as "eleven-years-old," this paper uses a word segmentation method based on phrase dictionary matching for text segmentation.

After word segmentation, we map the words to word vectors w_i^n according to word frequency. This paper sets $E_i = \{\text{Input}_1, \text{Input}_2, \ldots, \text{Input}_n\}$ to indicate that there are n tweets at a time, where $\text{Input}_i = \{w_i^1, w_i^2, \ldots, w_i^n\}$ means that the tweet has word vectors, and these word vectors are combined together and obtain the vector matrix e_i of the tweet formed after Input_i is processed by the words embedding layer.

3.2.2. Max-Pooling and Dropout Layer. In order to get the most prominent features of posts, the maximum pooling method is used for pooling, so that keywords or sentence features are reduced, and parameters are reduced. Finally, a fixed-size vector m_i can be generated. At the same time, in order to slow down overfitting and enhance the model generalization ability, the dropout layer is added.

3.2.3. BiGRU Layer. In order to strengthen the model's understanding of contextual semantics, this paper uses BiGRU to simultaneously combine the before and after sequences to make predictions. BiGRU is composed of two GRU stacked on top of each other, and its main structure is a combination of two unidirectional GRU. For each time t, the input will be provided to the two GRU in opposite directions at the same time, and the output will be jointly determined by the two unidirectional GRU.

As shown in Figure 3, x_t is the input data, h_t is the output of the GRU unit, z_t is the update gate, z_t and r_t jointly control the calculation from the hidden state of h_{t-1} to the hidden state of h_t, and the update gate also controls the current input data and previously memorized information h_{t-1}, output a value z_t between 0 and 1, and z_t determines how much h_{t-1} is transferred to the next state; the specific unit is calculated as the following formulae show:

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right), \tag{3}$$

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right), \tag{4}$$

$$\widetilde{h}_t = \tanh\left(W \cdot \left[r_t \times h_{t-1}, x_t\right]\right), \tag{5}$$

$$h_t = \left(1 - z_t\right) \times h_{t-1} + z_t \times \widetilde{h}_t, \tag{6}$$

where σ is the Sigmoid function and W_z, W_r, W are the weight matrix of update gate, reset gate, and candidate hidden state, respectively. The reset gate controls the importance of h_{t-1} to the result h_t. When the previous memory h_{t-1} is completely related to the new memory, the reset gate can be used to increase the impact of the previous memory. According to the calculation results of reset gate, update gate, and hidden state, the output h_t at the current moment can be obtained by formula (6), thereby obtaining the relationship between BiGRU and a large number of posts. Then, we use the final state h_N ($N =$ the number of posts received so far) to judge the rumors through the softmax function:

$$p = \text{softmax}\left(\omega^T \cdot \tanh\left(h_N\right)\right). \tag{7}$$

3.3. RL Model. In addition to the accuracy of detection, this paper also considers the timeliness of detection. This paper uses the Q-learning algorithm to dynamically determine the best checkpoint to improve the timeliness of rumor detection. The Q-learning algorithm has a calculation action value and a reward mechanism. The action value function calculates the Q value according to the obtained state expression. As shown in formula (8), combining the Q value and the post state expression, the action value function is used to determine whether to terminate or continue, and as shown in formula (9), the characteristic representations h_{fi} and h_{bi} are used as inputs to calculate the action value:

$$Q_{i+1}(s, a) = E\left[r + \gamma \max Q_i\left(s', a'\right) \mid s, a\right], \tag{8}$$

$$a_i = W_a\left(\text{ReLu}\left(W_h\left(h_{fi} + h_{bi}\right) + b_h\right)\right) + b_a, \tag{9}$$

where γ is the discount rate, r is the reward value, a^0 is rumor, and a^1 is nonrumor.

According to the action value and state value, we get the max reward value, used to optimize the action value.

The main features of the posts are maintained in the words embedding layer unchanged. Through the BiGRU layer, the new state value is obtained by combining the main features of the current input post and the previously obtained state value as the input of reinforcement learning. Then, the action value obtained by reinforcement learning decides whether to terminate or continue. If it terminates and the prediction is correct, the model will give a reward. If it terminates but the prediction is wrong, this model will give a big penalty. If it continues, this model will also give a small penalty. The calculation formula is shown in formula (10). In this way, the model will make a consideration between whether to continue or terminate in the end and, at the same time, make a trade-off between accuracy and timeliness:

$$\begin{cases} \log N & \text{terminate with correct prediction} \\ -P & \text{terminate with incorrect prediction} \\ -\varepsilon & \text{continue} \end{cases}, \tag{10}$$

where N is the number of correct predictions accumulated thus far, P is a large value to penalize an incorrect prediction, and ε is a small penalty value for delaying the detection.

4. Experiments and Analyses

4.1. Dataset. This paper uses the public standard Twitter dataset proposed by Ma et al. [18]. This dataset was proposed in 2016 and has been recognized by the academic community. It has since been widely used in the field of rumor detection and is a classic dataset on the problem of rumor detection. It contains 5802 events, each of which contains several Twitter-related JSON files. Each JSON file represents a tweet, including the information of the tweet and the basic information of the user who posted the tweet. The specific data information of the dataset is shown in Table 1. This paper uses part of the information to perform account filtering and then standardizes tweets, uses the creation time of the tweets as the test basis for early detection, encapsulates the data in a certain format, and uses the ratio of 8 : 2 to divide the preprocessed dataset into a training set and a test set for the application of the subsequent sequence in the training and testing of the model.

4.2. Evaluation Indicators. For the evaluation criteria of the model, this paper uses four indicators consistent with the literature [19, 20], namely, Accuracy, Precision, Recall, and $F1$. The calculation formula is shown as follows:

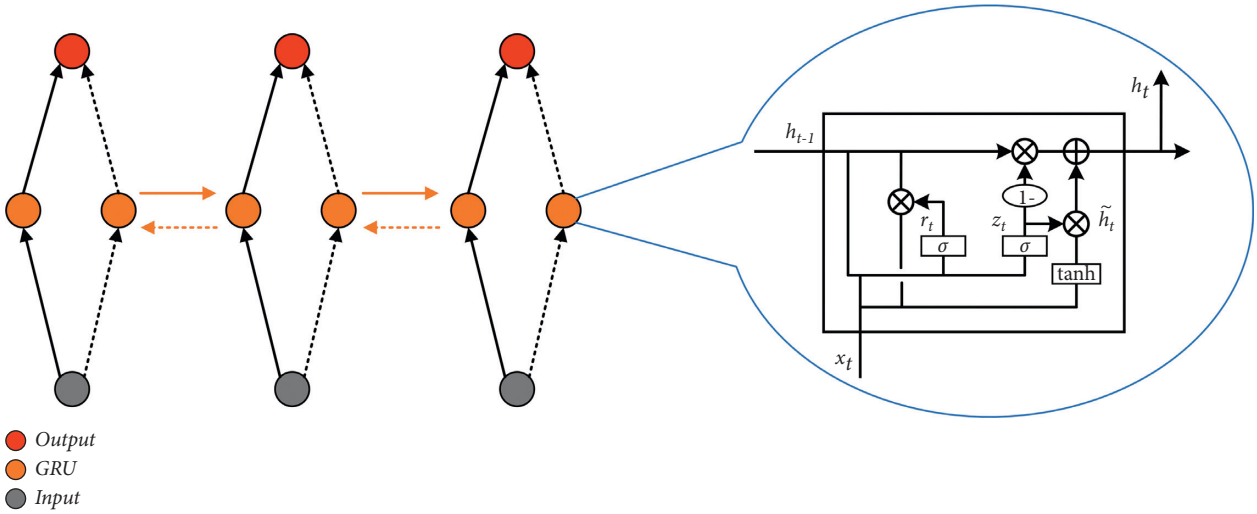


Figure 3: BiGRU (left) and GRU update process (right).

Table 1: Twitter standard dataset information.

Statistics	Twitter
User#	49,345
Posts#	103,212
Events#	5,802
Rumors#	1,972
Nonrumors	3,830
Avg. hours per event	33.4
Avg. # of posts per event	17
Max # of posts per event	346
Min # of posts per event	1

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}, \tag{11}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{12}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{13}$$

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{14}$$

where TP, FN, FP, TN are shown in Table 2 [21].

4.3. Environment and Hyperparameters. In this paper, the model is implemented under the Linux system, and the model is trained under the GPU environment of Python 3.6 and TensorFlow 1.13.1. In the training process, this paper uses the per-trained GloVe as the initialization of the word vector to input, which contains 840 billion words. Also, the dimensionality of word embedding is set to 300, while the dropout rate to embedding layer is 0.5. The Adam optimiser (Kingma and Ba) [22] with a learning rate of 0.01 for DL model and 0.001 for RL model are used as the optimization method. Set the size of each batch to 50 and the number of DL model and RL model alternate training rounds to 20.

Table 2: Confusion matrix.

		Actual	
		1	0
Predicted	1	True positive (TP)	False positive (FP)
	0	False negative (FN)	True negative (TN)

4.4. Results

4.4.1. Training Loss and Reward. The training loss and reward values over iterations are presented in Figures 4 and 5. In this paper, the DL model and RL model are trained in an alternating manner. It can be seen that the training loss of the DL model tends to be dynamically balanced after about 5000 iterations, and the loss value drops below 0.2, achieving the optimal value. In addition, the reward curve fluctuates more as the reward was calculated based on the accuracy of DL model. When switching between training DL and RL model, the reward value tends to change abruptly. But with the improvement of accuracy over time, a consistent improvement of reward value can be seen.

4.4.2. Detection Model and Comparison. In order to effectively testify the accuracy and effectiveness of the model, this paper compares the proposed model with the following models. The RNN model mainly uses recurrent neural networks to analyze data and detect whether a text is a rumor. The LSTM model is an improvement of the recurrent neural network model, which considers the contextual relationship. The GRU-2 model [18] is an improvement of the LSTM model, which reduces the parameters and improves the efficiency. Specifically, this model first divides the event into time periods, then uses the tf-idf method to calculate the text representation of each time period, uses a two-layer GRU network to learn the hidden layer representation of each event, and finally realizes the classification of the event. The HMM model [23] uses the group's point of view for analysis and finally achieves the classification effect. The GAN-GRU model [24] uses a generative adversarial network

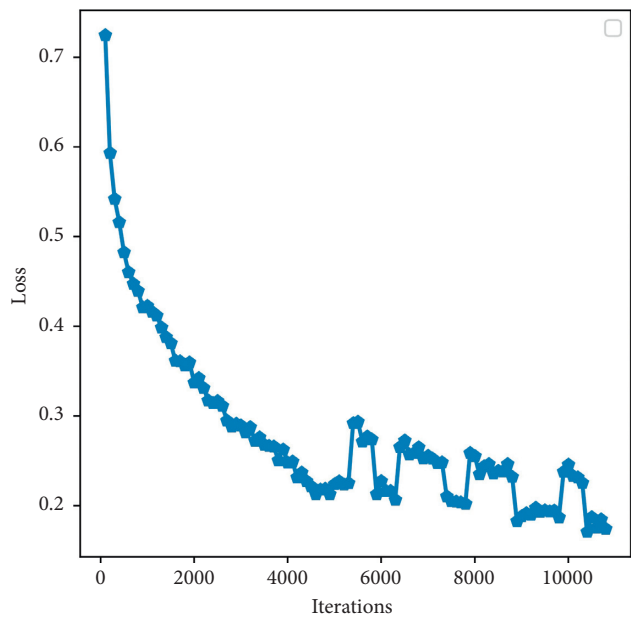


Figure 4: Changes in loss.

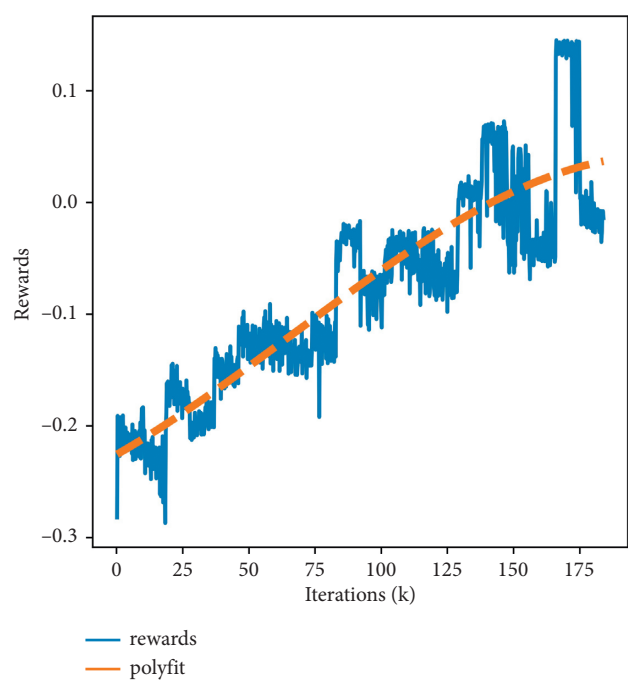


Figure 5: Reward during training.

to train the discriminator for rumor detection. The generator is mainly designed to produce uncertain or conflicting sounds, which complicates the original dialogue thread to make the different supercharged to learn stronger indicative representations of rumors from the enhanced and more challenging examples. The RDM model is a part of the ERD model. It mainly uses a single-layer GRU network to extract main information and uses reinforcement learning to conduct rumor detection in time.

In this section, this paper evaluates DDR based on four evaluation indicators on the test set, as shown in Table 3. This paper is mainly based on the improvement of the ERD model proposed in 2019 and proposes a preprocessing module based on account filtering and standardization. By processing the input of the model, the data is refined, and the accuracy of rumor detection is improved. It can be seen from the chart that the indicators of DDR are basically better than traditional models, such as RNN model, LSTM model,

Table 3: Detection accuracy on Twitter.

Method	Accuracy	Precision	Recall	$F1$
Baseline	0.612	0.355	0.465	0.398
RNN	0.785	0.707	0.659	0.682
LSTM	0.796	0.719	0.683	0.701
GRU-2	0.808	0.741	0.694	0.717
HMM	—	—	—	0.524
GAN-GRU	0.781	0.773	0.796	0.784
RDM	**0.873**	0.817	0.823	0.820
ERD	0.858	**0.843**	0.735	0.785
DDR (ours)	**0.863**	0.816	**0.858**	**0.836**

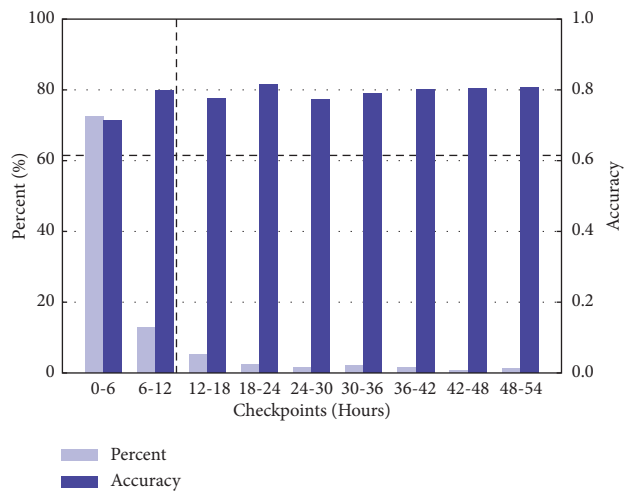


Figure 6: Proportion of Twitter events classified and detection accuracy by DDR over time. Vertical dashed line indicates the optimal checkpoint for GRU-2 and horizontal dotted line indicates GRU-2's accuracies.

HMM model, and GAN-GRU model. At the same time, the model in this paper is compared with the ERD model that also considers the timeliness of rumor detection. The ERD model has an accuracy of 0.858 in the accuracy of detection. After improving on the basis of the ERD, the accuracy of the model has reached 0.863, an increase of 0.005.

Comparing the three indicators of Precision, Recall, and $F1$, the model proposed in this paper is 0.027 lower than the ERD model on the Precision indicator, and the other two indicators have increased by 0.123 and 0.051, respectively. It can be seen that the model proposed in this paper has a good effect in the process of rumor detection. However, when reinforcement learning is added, reinforcement learning must consider the timeliness of detection, and you will want to get results faster when making trade-offs. Therefore, reinforcement learning has an impact on the accuracy of the model, compared to not considering timeliness. For the RDM model, the accuracy of the model proposed in this paper is slightly lower than 0.01.

In summary, the model proposed in this paper improves the accuracy of rumor detection while considering the timeliness of model detection. It can be seen from the graph that the proposed model is effective in detecting rumors.

4.4.3. Detection Timeliness. Then, in order to evaluate the timeliness of detection, based on the Twitter standard dataset, this paper focuses on comparing the DDR model with the GRU-2 model. The biggest difference between the two models is that GRU-2 uses fixed checkpoints. The DDR dynamically determines checkpoints through the DL model.

This paper presents the proportion of events that are classified by DDR and the classification accuracy over time (6-hour interval) in Figure 6. Firstly, it can be seen that 70% of rumors are discovered within 6 hours, and the best checkpoint of GRU-2 (vertical dashed line) is 12 hours, so DDR can detect most rumors earlier than GRU-2. Secondly, it can be clearly seen that the classification accuracy of DDR is better than GRU-2 (horizontal dotted line) at all checkpoints.

In summary, it can be seen that DDR improves the timeliness of rumor detection compared with the GRU-2 model.

5. Conclusions

This paper presents a new early rumor detection model. The model is divided into three submodels: DP model, DL model, and RL model. In rumor detection, according to the early rumor detection model, this paper proposes to filter the account according to the user's portrait characteristics and the user tweet's praise number. The text standardization for tweets is defined. The original data is processed more precisely and more precisely to improve the accuracy of the detection. At the same time, BiGRU is proposed to enhance

the understanding of context semantics, fully consider the relationship between the post text, and improve the effect of model training. The model is trained by the method of intensive learning to ensure the timeliness of rumor detection. The results are compared with the early rumor detection model. The results show that the accuracy of DDR model is 86.3% in the test set, 0.5% higher than ERD model, and 5.5% higher than that of GRU-2 model. It is proved that the model proposed in this paper has achieved good results in rumor detection. 70% of them are found in 6 hours, and the detection of GRU-2 model takes 12 hours. Therefore, this paper improves the accuracy of rumor discovery on the premise of ensuring the timeliness of rumor discovery. With the development of science and technology, there is an increasing trend about the investigation of the real world problems such as the development of Faults Detection [25], Vortex Theory [26], Periodic Orbit [27], Dynamics [28] and other research fields. It is hoped that the model proposed in this paper can be applied to these problems in the near future.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Tianjin under Grant no. 19JCYBJC15400 and the Program for the Science and Technology Plans of Tianjin, China, under Grant 20YDTPJC01200.

References

[1] J. Ma, W. Gao, Z. Wei et al., "Detect rumors using time series of social context information on microblogging websites," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, vol. 3, pp. 1751–1754, Melbourne Australia, October 2015.

[2] S. Kwon, M. Cha, and K. Jung, "Rumor detection over varying time windows," *PLoS One*, vol. 12, no. 1, pp. e0168344–19, 2017.

[3] M. Farajtabar, J. Yang, X. Ye et al., "Fake news mitigation via point process based intervention," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 1097–1106, Sydney, Australia, August 2017.

[4] C. Liang, Y. Qiu, and S. Lu, "Research on microblog rumor detection method," *Computer applications and software*, vol. 30, no. 2, pp. 226–228, 2013.

[5] T. Lu, B. Shi, Z. Yan et al., "A semi supervised learning algorithm for microblog rumor detection," *Computer application research*, vol. 33, no. 3, pp. 744–748, 2016.

[6] E. Mao, G. Chen, X. Liu et al., "Research on microblog rumor detection based on deep feature and ensemble classifier," *Computer application research*, vol. 33, no. 11, pp. 3369–3373, 2016.

[7] T. Takahashi and N. Igata, "Rumor detection on twitter," in *Proceedings of the 6th International Conference On Soft Computing and Intelligent Systems*, pp. 452–457, Kobe, Japan, November 2012.

[8] N. Karamchandani and M. Franceschetti, "Rumor source detection under probabilistic sampling," in *Proceedings of the IEEE International Symposium On Information Theory Proceedings (ISIT)*, Istanbul, Turkey, July 2013.

[9] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina Weibo by propagation structures," in *Proceedings of the IEEE 31st International Conference On Data Engineering (ICDE)*, pp. 651–662, Seoul, South Korea, April 2015.

[10] L. Li, G. Cai, and J. pan, "Microblog rumor detection method based on C-GRU," *Journal of Shandong University*, vol. 49, no. 2, pp. 102–106, 2019.

[11] W. Ren, B. Qin, and T. Liu, "Research on rumor detection based on time series network," *Intelligent computer and application*, vol. 9, no. 3, pp. 300–303, 2019.

[12] X. Liao, Z. Huang, D. Yang et al., "Social media rumor detection based on hierarchical attention network," *Chinese Science: Information Science*, vol. 48, no. 11, pp. 1558–1574, 2018.

[13] S. Srinivasan and L. D. Dhinesh Babu, "A parallel neural network approach for faster rumor identification in online social networks," *International Journal on Semantic Web and Information Systems*, vol. 15, no. 4, pp. 69–89, 2019.

[14] K. Zhou, S. Chang, B. Li et al., "Early rumour detection," *Association for Computational Linguistics*, vol. 1, pp. 1614–1623, 2019.

[15] D. Z. Lin, B. Ma, D. L. Cao, and J. H. Lau, "Chinese microblog rumor detection based on deep sequence context," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 23, 2019.

[16] M. Z. Asghar, A. Habib, A. Habib et al., "Exploring deep neural networks for rumor detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, 2019.

[17] S. M. Alzanin and A. M. Azmi, "Rumor detection in arabic tweets using semi-supervised and unsupervised expectation-maximization," *Knowledge-Based Systems*, vol. 185, 2019.

[18] J. Ma, W. Gao, P. Mitra et al., "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 3818–3824, New York, NY, USA, July 2016.

[19] F. Yu, Q. Liu, S. Wu et al., "A convolutional approach for misinformation identification," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3901–3907, Melbourne, Australia, August 2017.

[20] N. Ruchansky, S. Seo, and Y. Liu, "CSI: a hybrid deep model for fake news detection," in *Proceedings of the ACM on Conference on Information and Knowledge Management*, pp. 797–806, ACM, Singapore, Asia, November 2017.

[21] M. Yuan and H. Tang, "Research on rumor identification based on graph convolution network," *Computer Engineering and Application*, vol. 57, no. 13, 2020.

[22] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[23] S. Dungs, A. Aker, N. Fuhr et al., "Can rumour stance alone predict veracity," in *Proceedings of the 27th Inter-national Conference on Computational Linguistics*, pp. 3360–3370, Santa Fe, NM, USA, August 2018.

[24] J. Ma, W. Gao, and K. F. Wong, "Detect rumors on twitter by promoting information campaigns with generative adversarial learning," in *Proceedings of the World Wide Web Conference*, pp. 3049–3055, San Francisco, CA, USA, May 2019.

[25] M. Miguel, J. A. Antonino-Daviu, F. D. C. Castellá Pedro, and C. Alberto, "Higher-order spectral analysis of stray flux signals for faults detection in induction motors," *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 2, pp. 1–14, 2020.

[26] M. Sharifi and B. Raesi, "Vortex theory for two dimensional Boussinesq equations," *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 2, pp. 67–84, 2020.

[27] A. A. Abozaid, H. H. Selim, K. A. K. Gadallah, I. A. Hassan, and E. I. Abouelmagd, "Periodic orbit in the frame work of restricted three bodies under the asteroids belt effect," *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 2, pp. 157–176, 2020.

[28] S. S. Hassan, M. P. Reddy, and R. K. Rout, "Dynamics of the modified n-degree Lorenz system," *Applied Mathematics and Nonlinear Sciences*, vol. 4, no. 2, pp. 315–330, 2019.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 9688627, 17 pages
https://doi.org/10.1155/2021/9688627

Research Article

Passivity Analysis of Coupled Stochastic Neural Networks with Multiweights

Min Cao,[1] **Xun-Wu Yin**,[2] **Wen-He Song**,[3] **Xue-Mei Sun**,[1] **Cheng-Dong Yang**,[4] **and Shun-Yan Ren**[5]

[1]*Tianjin Key Laboratory of Autonomous Intelligences Technology and System, School of Computer Science and Technology, Tiangong University, Tianjin 300387, China*
[2]*School of Mathematical Sciences, Tiangong University, Tianjin 300387, China*
[3]*Academy of Science and Technology, Tiangong University, Tianjin 300387, China*
[4]*School of Information Science and Technology, Linyi University, Linyi 276005, China*
[5]*School of Mechanical Engineering, Tiangong University, Tianjin 300387, China*

Correspondence should be addressed to Cheng-Dong Yang; yangchengdong@lyu.edu.cn and Shun-Yan Ren; renshunyan@163.com

Received 19 May 2021; Revised 3 July 2021; Accepted 12 July 2021; Published 3 August 2021

Academic Editor: Ya Jia

In this paper, we devote to the investigation of passivity in two types of coupled stochastic neural networks (CSNNs) with multiweights and incompatible input and output dimensions. First, some new definitions of passivity are proposed for stochastic systems that may have incompatible input and output dimensions. By utilizing stochastic analysis techniques and Lyapunov functional method, several sufficient conditions are respectively developed for ensuring that CSNNs without and with multiple delay couplings can realize passivity. Besides, the synchronization criteria for CSNNs with multiweights are established by employing the results of output-strictly passivity. Finally, two simulation examples are given to illustrate the validity of the theoretical results.

1. Introduction

In recent decades, neural networks (NNs) have potential applications in the image encryption, pseudorandom number generators, optimization, and other areas [1–3], which depend on the dynamical behaviors of NNs including stability and passivity. Therefore, the stability [4–8] and passivity [9–14] for various NNs have received special attention in recent years. Mou et al. [4] considered the asymptotic stability problem for Hopfield NNs with time delay via combining the Lyapunov functional and delay fractioning approach. Yang et al. [6] discussed the stability for a kind of NNs with time-varying delays and gave several delay-dependent stability conditions by taking advantage of integral inequality. In [9], a class of NNs with time-varying delays and parameter uncertainties was took into account, and some exponential passivity criteria were established by exploiting weighted integral inequalities. Xiao et al. [11] studied the passivity for a type of memristive NNs with inertial term, obtained some criteria of asymptotic stability by utilizing the passivity, and discussed the case that parameters are uncertain but bounded.

As it is known to all, stochastic perturbations are unavoidable in the implementation of NNs and may cause undesirable dynamical behaviors in NNs [15, 16]. Therefore, the dynamical behaviors including the stability [17–21] and passivity [22–27] have been widely investigated by numerous researchers for NNs with stochastic perturbations in recent years. In [18], several sufficient conditions on mean square stability for stochastic neural networks (SNNs) with local impulses were derived by using the mathematical induction method. Yang and Li [20] coped with the stability problem for switched SNNs with parameter uncertainties, and derived several conditions to guarantee the robust

stability by utilizing the state-dependent switching method. In [24], the authors took into account one type of uncertain SNNs with distributed and discrete time-varying delays and gave some passivity criteria with the help of integral inequality technique. Nagamani et al. [25], respectively, discussed the passivity and dissipativity for Markovian jump stochastic NNs with two types of time-varying delays and obtained several delay-dependent passivity and dissipativity criteria by taking a suitable Lyapunov functional.

Coupled neural networks (CNNs) comprised of a number of NNs have tremendous potential applications in many areas of engineering [28–30]. Hence, the dynamical behaviors of CNNs have attracted much attention; especially, the passivity [31–35] and synchronization [36–40] for many types of CNNs have been deeply discussed. In [34], the authors not only obtained several passivity criteria for the directed CNNs based on the developed adaptive control strategies but also discussed the case that topologies are undirected. Qi et al. [37] derived several global exponential synchronization criteria for quaternion-valued CNNs with impulses by exploiting Lyapunov functional and matrix inequalities. In [38], the authors considered CNNs with mixed impulses and developed some exponential synchronization conditions for the proposed network model through using delayed impulsive differential inequality. Particularly, a number of authors have discussed the dynamical behavior of coupled SNNs (CSNNs) in recent years [41–45]. Chen et al. [44] utilized the adaptive feedback controller to deal with the exponential synchronization for CSNNs. In [45], the authors respectively employed time-triggered and event-triggered impulsive control methods to investigate the synchronization of discrete time CSNNs with multidelays. Unfortunately, the passivity of CSNNs has not yet been investigated.

It should be pointed out that the results in [31–45] only focused on CNNs with single weight. Considering the diversity of influencing factors, some researchers discussed the synchronization and passivity for multiweighted CNNs (MWCNNs) [46–48]. In [46], the authors proposed MWCNNs without and with coupling delays and dealt with the passivity and synchronization for these network models by employing the impulsive control method. Wang and Zhao [48] not only discussed the passivity for MWCNNs by utilizing the designed proportional-integral and proportional-derivative controllers but also derived several synchronization criteria by virtue of output-strict passivity. However, the passivity and synchronization for multiweighted CSNNs (MWCSNNs) have not been investigated.

In this paper, the passivity for two types of MWCSNNs with incompatible input and output dimensions is investigated. The main contributions have three aspects. First, we present several new definitions of passivity for stochastic systems with incompatible dimensions of output and input. Second, some sufficient conditions to ensure the passivity of MWCSNNs are obtained by taking advantage of the Lyapunov functional method and stochastic analysis techniques, and a synchronization criterion is also developed by utilizing the result of output-strictly passivity. Third, we further address the passivity and synchronization for CSNNs with multiple delay couplings (CSNNMDCs).

2. Preliminaries

2.1. Notations. $(\Omega, \mathscr{F}, \{\mathscr{F}_t\}_{t\geq 0}, \mathscr{P})$ is a complete probability space with the natural filtration $\{\mathscr{F}_t\}_{t\geq 0}$ satisfying the usual conditions. $C^{1,2}(\mathbb{R}^+ \times \mathbb{R}^n; \mathbb{R}^+)$ represents the family of all nonnegative functions $V(t, \kappa(t))$ on $\mathbb{R}^+ \times \mathbb{R}^n$, which are once differentiable in t and twice continuously differentiable in $\kappa(t)$. $\mathrm{Tr}(\cdot)$ stands for the trace of a matrix. $P \geq 0$ is used to denote a symmetric semipositive definite matrix. $\lambda_M(\cdot)$ and $\lambda_m(\cdot)$, respectively, denote the maximum and minimum eigenvalue of a real symmetric matrix.

2.2. Lemmas

Lemma 1 *(Itô formula, see [49]). A stochastic system can be described by*

$$\mathrm{d}\kappa(t) = f(t, \kappa(t))\mathrm{d}t + g(t, \kappa(t))\mathrm{d}w(t), \tag{1}$$

in which $\kappa(t) \in \mathbb{R}^n$ represents the state of system, $f(\cdot)$: $\mathbb{R}^+ \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is continuous nonlinear function, $g(\cdot)$: $\mathbb{R}^+ \times \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times n}$ is noise intensity function, and $w(t)$ is an n-dimensional Brownian motion (Wiener process) defined on $(\Omega, \mathscr{F}, \{\mathscr{F}_t\}_{t\geq 0}, \mathscr{P})$.

For any $V(t, \kappa(t)) \in C^{1,2}(\mathbb{R}^+ \times \mathbb{R}^n; \mathbb{R}^+)$, define the operator $\mathscr{L}V$ for system (1) as follows:

$$\begin{aligned}\mathscr{L}V(t, \kappa(t)) &= V_t(t, \kappa(t)) + V_\kappa(t, \kappa(t))f(t, \kappa(t)) \\ &\quad + \frac{1}{2}\mathrm{Tr}\left(g^T(t, \kappa(t))V_{\kappa\kappa}(t, \kappa(t))g(t, \kappa(t))\right),\end{aligned} \tag{2}$$

where $V_t(t, \kappa(t)) = (\partial V(t, \kappa(t))/\partial t)$, $V_\kappa(t, \kappa(t)) = ((\partial V(t, \kappa(t))/\partial\kappa_1), (\partial V(t, \kappa(t))/\partial\kappa_2), \ldots, (\partial V(t, \kappa(t))/\partial\kappa_n))$, $V_{\kappa\kappa}(t, \kappa(t)) = (\partial^2 V(t, \kappa(t))/\partial\kappa_i\kappa_j)_{n\times n}$.

If $V(t, \kappa(t)) \in C^{1,2}(\mathbb{R}^+ \times \mathbb{R}^n; \mathbb{R}^+)$, one has

$$\mathbb{E}V(t, \kappa(t)) = \mathbb{E}V(t_0, \kappa(t)) + \mathbb{E}\left\{\int_{t_0}^{t} \mathscr{L}V(t, \kappa(t))\mathrm{d}t\right\}, \tag{3}$$

for all $t > t_0 \geq 0$.

Lemma 2 (see [50]). *For any matrices $M \in \mathbb{R}^{m\times n}$ and $0 \leq P \in \mathbb{R}^{m\times m}$, one obtains*

$$\mathrm{Tr}\left(M^T P M\right) \leq \lambda_M(P)\mathrm{Tr}\left(M^T M\right). \tag{4}$$

2.3. Definitions

Definition 1. A stochastic system with input $\beta(t) \in \mathbb{R}^p$ and output $\eta(t) \in \mathbb{R}^q$ is passive if

$$\mathbb{E}\left\{\int_{\vartheta_0}^{\vartheta_p} \eta^T(s)F\beta(s)\mathrm{d}s\right\} \geq \mathbb{E}\left\{S\left(\vartheta_p\right)\right\} - \mathbb{E}\{S(\vartheta_0)\}, \tag{5}$$

for any $\vartheta_p, \vartheta_0 \in \mathbb{R}^+$ and $\vartheta_p \geq \vartheta_0$, in which $F \in \mathbb{R}^{q\times p}$ and S is a nonnegative function.

Definition 2. A stochastic system with input $\beta(t) \in \mathbb{R}^p$ and output $\eta(t) \in \mathbb{R}^q$ is input-strictly passive if

$$\mathbb{E}\left\{\int_{\vartheta_0}^{\vartheta_p} \eta^T(s)F\beta(s)\mathrm{d}s\right\} \geq \mathbb{E}\left\{S\left(\vartheta_p\right)\right\} - \mathbb{E}\{S(\vartheta_0)\} + \mathbb{E}\left\{\int_{\vartheta_0}^{\vartheta_p} \beta^T(s)A_1\beta(s)\mathrm{d}s\right\}, \tag{6}$$

for any $\vartheta_p, \vartheta_0 \in \mathbb{R}^+$ and $\vartheta_p \geq \vartheta_0$, in which $F \in \mathbb{R}^{q\times p}, 0 < A_1 \in \mathbb{R}^{p\times p}$, and S is a nonnegative function.

Definition 3. A stochastic system input $\beta(t) \in \mathbb{R}^p$ and output $\eta(t) \in \mathbb{R}^q$ is output-strictly passive if

$$\mathbb{E}\left\{\int_{\vartheta_0}^{\vartheta_p} \eta^T(s)F\beta(s)\mathrm{d}s\right\} \geq \mathbb{E}\left\{S\left(\vartheta_p\right)\right\} - \mathbb{E}\{S(\vartheta_0)\} + \mathbb{E}\left\{\int_{\vartheta_0}^{\vartheta_p} \eta^T(s)A_2\eta(s)\mathrm{d}s\right\}, \tag{7}$$

for any $\vartheta_p, \vartheta_0 \in \mathbb{R}^+$ and $\vartheta_p \geq \vartheta_0$, in which $F \in \mathbb{R}^{q\times p}, 0 < A_2 \in \mathbb{R}^{q\times q}$, and S is a nonnegative function.

3. Passivity for MWCSNNs

3.1. Network Model. The MWCSNNs in this paper is considered as follows:

$$\begin{aligned}\mathrm{d}\kappa_z(t) = &\left(-D\kappa_z(t) + Gf\left(\kappa_z(t)\right) + B + \sum_{n=1}^{s}\sum_{h=1}^{M} b_n C_{zh}^n \Gamma^n \kappa_h(t) + H\beta_z(t)\right)\mathrm{d}t \\ &+ \sigma\left(\kappa_z(t)\right)\mathrm{d}\omega(t), \quad z = 1, 2, \ldots, M,\end{aligned} \tag{8}$$

where $n = 1, 2, \ldots, s$, $G = (G_{zj})_{m\times m} \in \mathbb{R}^{m\times m}, G_{zj} \in \mathbb{R}$, represents the strength of the j th neuron on the z th neuron, $H \in \mathbb{R}^{m\times p}$ is the known matrix, $\kappa_z(t) = (\kappa_{z1}(t), \kappa_{z2}(t), \ldots, \kappa_{zm}(t))^T \in \mathbb{R}^m$ is the state vector of the z th node, $\beta_z(t) \in \mathbb{R}^p$ represents the external input, $0 < D = \mathrm{diag}(d_1, d_2, \ldots, d_m) \in \mathbb{R}^{m\times m}$, $0 < d_z \in \mathbb{R}$, represents the rate with which the z th neuron will reset its potential to the resting state when disconnected from the network and external input, $B = (B_1, B_2, \ldots, B_m)^T \in \mathbb{R}^m$, $f(\kappa_z(t)) = (f_1(\kappa_{z1}(t)), f_2(\kappa_{z2}(t)), \ldots, f_m(\kappa_{zm}(t)))^T \in \mathbb{R}^m$, $0 < b_n \in \mathbb{R}$, denotes coupling strength, and $C^n = (C_{zh}^n)_{M\times M} \in \mathbb{R}^{M\times M}$ represents the outer coupling matrix, where C_{zh}^n satisfies the following conditions: if there exists a connection between nodes z and h, then $\mathbb{R} \ni C_{zh}^n = C_{hz}^n > 0\,(z \neq h)$, otherwise, $\mathbb{R} \ni C_{zh}^n = C_{hz}^n = 0$, and $C_{zz}^n = -\sum_{\substack{h=1\\h\neq z}}^{M} C_{zh}^n$, $z = 1, 2, \ldots, M$; $0 < \Gamma^n = \mathrm{diag}(\Gamma_1^n, \Gamma_2^n, \ldots, \Gamma_m^n) \in \mathbb{R}^{m\times m}$ denotes the inner coupling matrix; $\omega(t) = (\omega_1(t), \omega_2(t), \ldots, \omega_m(t))^T \in \mathbb{R}^m$ is an m-dimensional Brownian motion defined on $(\Omega, \mathscr{F}, \{\mathscr{F}_t\}_{t\geq 0}, \mathscr{P})$; $\sigma(\kappa_z(t)) = \mathrm{diag}(\sigma(\kappa_{z1}(t)), \sigma(\kappa_{z2}(t)), \ldots, \sigma(\kappa_{zm}(t))) \in \mathbb{R}^{m\times m}$ is the noise intensity matrix.

In this paper, the following assumptions are made.

Assumption 1. $f_l(\cdot)\,(l = 1, 2, \ldots, m)$ satisfies

$$\left|f_l(j_1) - f_l(j_2)\right| \leq \xi_l\left|j_1 - j_2\right|, \tag{9}$$

for any $j_1, j_2 \in \mathbb{R}$, where $\xi_l > 0$. Let $\xi = \mathrm{diag}(\xi_1^2, \xi_2^2, \ldots, \xi_m^2) \in \mathbb{R}^{m\times m}$.

Assumption 2. There exists a positive constant μ such that $\sigma(\cdot)$ satisfies the following inequality:

$$\mathrm{Tr}\left[\left(\sigma(k_1) - \sigma(k_2)\right)^T\left(\sigma(k_1) - \sigma(k_2)\right)\right] \leq \mu^2\left\|k_1 - k_2\right\|^2, \tag{10}$$

for any $k_1, k_2 \in \mathbb{R}^n$.

Remark 1. On the one hand, the passivity for various CNNs has been investigated and some meaningful results have been obtained [31–35]. However, the passivity of CSNNs has not yet been discussed. On the other hand, some researchers have dealt with the synchronization problem for CSNNs [41–45]. Given that passivity has been developed as a powerful tool to solve the synchronization problem of CNNs, the investigation on synchronization for CSNNs based on the passivity is apparently very valuable. Regrettably, the result about this topic has not yet been reported.

3.2. Passivity Analysis. Suppose that $s(t) = (s_1(t), s_2(t), \ldots, s_m(t))^T \in \mathbb{R}^m$ is an arbitrary desired solution of the isolated node of system (8), then it satisfies

$$\mathrm{d}s(t) = (-Ds(t) + Gf(s(t)) + B)\mathrm{d}t + \sigma(s(t))\mathrm{d}\omega(t). \tag{11}$$

Letting $\alpha_z(t) = (\alpha_{z1}(t), \alpha_{z2}(t), \ldots, \alpha_{zm}(t))^T = \kappa_z(t) - s(t)$, we can obtain from (8) and (11) that

$$\begin{aligned}\mathrm{d}\alpha_z(t) = &\left(-D\alpha_z(t) + Gf(\kappa_z(t)) - Gf(s(t)) + \sum_{n=1}^{s}\sum_{h=1}^{M} b_n C_{zh}^n \Gamma^n(\kappa_h(t) - s(t)) + H\beta_z(t)\right)\mathrm{d}t \\ &+ (\sigma(\kappa_z(t)) - \sigma(s(t)))\mathrm{d}\omega(t).\end{aligned} \tag{12}$$

The output vector $\eta_z(t) \in \mathbb{R}^q$ of network (12) is defined as follows:

$$\eta_z(t) = Z_1\alpha_z(t) + Z_2\beta_z(t), \tag{13}$$

where $Z_1 \in \mathbb{R}^{q\times m}$ and $Z_2 \in \mathbb{R}^{q\times p}$.

Define $\widehat{D} = \mathrm{diag}(D, D, \ldots, D) \in \mathbb{R}^{Mm\times Mm}$, $\widehat{G} = \mathrm{diag}(G, G, \ldots, G) \in \mathbb{R}^{Mm\times Mm}$, $\widehat{H} = \mathrm{diag}(H, H, \ldots, H) \in \mathbb{R}^{Mm\times Mp}$, $\widehat{Z}_1 = \mathrm{diag}(Z_1, Z_1, \ldots, Z_1) \in \mathbb{R}^{Mq\times Mm}$, $\widehat{Z}_2 = \mathrm{diag}(Z_2, Z_2, \ldots, Z_2) \in \mathbb{R}^{Mq\times Mp}$, $\widehat{\xi} = \mathrm{diag}(\xi, \xi, \ldots, \xi) \in \mathbb{R}^{Mm\times Mm}$, and $\widehat{\omega}(t) = (\omega^T(t), \omega^T(t), \ldots, \omega^T(t))^T \in \mathbb{R}^{Mm}$.

Take

$$\begin{aligned}\alpha(t) &= \left(\alpha_1^T(t), \alpha_2^T(t), \ldots, \alpha_M^T(t)\right)^T, \\ \beta(t) &= \left(\beta_1^T(t), \beta_2^T(t), \ldots, \beta_M^T(t)\right)^T, \\ \eta(t) &= \left(\eta_1^T(t), \eta_2^T(t), \ldots, \eta_M^T(t)\right)^T.\end{aligned} \tag{14}$$

According to (12), we have

$$\begin{aligned}\mathrm{d}\alpha(t) &= \left(-\widehat{D}\alpha(t) + \widehat{G}(f(\kappa(t)) - f(s(t))) + \widehat{H}\beta(t) + \sum_{n=1}^{s} b_n(C^n \otimes \Gamma^n)\alpha(t)\right)\mathrm{d}t + \widehat{\sigma}(t)\mathrm{d}\widehat{\omega}(t), \\ \eta(t) &= \widehat{Z}_1\alpha(t) + \widehat{Z}_2\beta(t),\end{aligned} \tag{15}$$

where $\widehat{\sigma}(t) = I_M \otimes (\sigma(\kappa(t)) - \sigma(s(t)))$.

Theorem 1. *Network (15) is passive if there exist $F \in \mathbb{R}^{Mq\times Mp}$, and $0 < \widehat{P} \in \mathbb{R}^{Mm\times Mm}$ satisfying*

$$\begin{pmatrix} W_1 & E_1 \\ E_1^T & M_1 \end{pmatrix} \le 0, \tag{16}$$

where $W_1 = -\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + q\,\mu^2 I_{Mm} + \sum_{n=1}^{s} b_n \widehat{P}(C^n \otimes \Gamma^n) + \sum_{n=1}^{s} b_n (C^n \otimes \Gamma^n)\widehat{P}$, $E_1 = \widehat{P}\widehat{H} - (\widehat{Z}_1^T F)/2$, $M_1 = -(\widehat{Z}_2^T F + F^T \widehat{Z}_2)/2$, and $q = \lambda_M(\widehat{P})$.

Proof. For convenience, we denote

$$y(t) = -\widehat{D}\alpha(t) + \widehat{G}(f(\kappa(t)) - f(s(t))) + \widehat{H}\beta(t) + \sum_{n=1}^{s} b_n(C^n \otimes \Gamma^n)\alpha(t), \tag{17}$$

then network (15) can be rewritten as

$$\mathrm{d}\alpha(t) = y(t)\mathrm{d}t + \widehat{\sigma}(t)\mathrm{d}\widehat{\omega}(t). \tag{18}$$

Choose the following Lyapunov functional for network (15):

$$V^1(t) = \alpha^T(t)\widehat{P}\alpha(t). \tag{19}$$

In light of Lemma 1, we can obtain

$$\begin{aligned}\mathscr{L}V^1(t) &= V_t^1(t) + V_\alpha^1(t)y(t) + \frac{1}{2}\mathrm{Tr}\left(\widehat{\sigma}^T(t)V_{\alpha\alpha}^1(t)\widehat{\sigma}(t)\right)\\ &= 2\alpha^T(t)\widehat{P}\left(-\widehat{D}\alpha(t) + \widehat{G}(f(\kappa(t)) - f(s(t))) + \widehat{H}\beta(t) + \sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\alpha(t)\right) + \mathrm{Tr}\left(\widehat{\sigma}^T(t)\widehat{P}\widehat{\sigma}(t)\right).\end{aligned} \tag{20}$$

Obviously,

$$2\alpha^T(t)\widehat{P}\widehat{G}(f(\kappa(t)) - f(s(t))) \le \alpha^T(t)\left(\widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi}\right)\alpha(t). \tag{21}$$

From Lemma 2 and Assumption 2, we have

$$\begin{aligned}&\mathrm{Tr}\left(\widehat{\sigma}^T(t)\widehat{P}\widehat{\sigma}(t)\right)\\ &\le q\sum_{z=1}^{M}\mathrm{Tr}\left(\left(\sigma(\kappa_z(t)) - \sigma(s(t))\right)^T\left(\sigma(\kappa_z(t)) - \sigma(s(t))\right)\right)\\ &\le q\sum_{z=1}^{M}\mu^2\left\|\kappa_z(t) - s(t)\right\|^2\\ &= q\mu^2\|\alpha(t)\|^2.\end{aligned} \tag{22}$$

Furthermore,

$$\begin{aligned}&2\alpha^T(t)\widehat{P}\left(\sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\alpha(t)\right)\\ &= \alpha^T(t)\left(\sum_{n=1}^{s} b_n\widehat{P}(C^n\otimes\Gamma^n) + \sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\widehat{P}\right)\alpha(t).\end{aligned} \tag{23}$$

By (20)–(23), one obtains

$$\begin{aligned}\mathscr{L}V^1(t) \le{}& \alpha^T(t)\left(-\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + q\mu^2 I_{Mm} + \sum_{n=1}^{s} b_n\widehat{P}(C^n\otimes\Gamma^n) + \sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\widehat{P}\right)\alpha(t)\\ &+ 2\alpha^T(t)\widehat{P}\widehat{H}\beta(t).\end{aligned} \tag{24}$$

From (24), we have

$$\begin{aligned}\mathscr{L}V^1(t) - \eta^T(t)F\beta(t) \le{}& \alpha^T(t)\left(-\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + q\mu^2 I_{Mm}\right.\\ &\left.+ \sum_{n=1}^{s} b_n\widehat{P}(C^n\otimes\Gamma^n) + \sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\widehat{P}\right)\\ \alpha(t) + 2\alpha^T(t)\widehat{P}\widehat{H}\beta(t) - \left(\alpha^T(t)\widehat{Z}_1^T + \beta^T(t)\widehat{Z}_2^T\right)F\beta(t) &= \zeta^T(t)\begin{pmatrix} W_1 & E_1\\ E_1^T & M_1\end{pmatrix}\zeta(t),\end{aligned} \tag{25}$$

where $\zeta(t) = (\alpha^T(t), \beta(t)^T)^T$.

By (16), one obtains

$$\eta^T(t)F\beta(t) \ge \mathscr{L}V^1(t). \tag{26}$$

From (26), we obtain

$$\int_{t_0}^{t_p}\eta^T(t)F\beta(t)\mathrm{d}t \ge \int_{t_0}^{t_p}\mathscr{L}V^1(t)\mathrm{d}t. \tag{27}$$

According to (27) and Lemma 1, we can acquire

$$\mathbb{E}\left\{\int_{t_0}^{t_p}\eta^T(t)F\beta(t)\mathrm{d}t\right\} \ge \mathbb{E}\left\{V^1(t_p)\right\} - \mathbb{E}\left\{V^1(t_0)\right\}. \tag{28}$$

□

Theorem 2. *Network (15) is input-strictly passive if there exist matrices* $F \in \mathbb{R}^{Mq\times Mp}, 0 < A_1 \in \mathbb{R}^{Mp\times Mp}$, *and* $0 < \widehat{P} \in \mathbb{R}^{Mm\times Mm}$ *satisfying*

$$\begin{pmatrix} W_1 & E_1 \\ E_1^T & M_2 \end{pmatrix} \le 0, \tag{29}$$

where W_1 *and* E_1 *have the same meanings as these in Theorem 1 and* $M_2 = M_1 + A_1$.

Proof. Construct the same $V^1(t)$ as (19) for network (15), and we can easily obtain

$$\begin{aligned}
&\mathscr{L}V^1(t) - \eta^T(t)F\beta(t) + \beta^T(t)A_1\beta(t) \\
&\le \alpha^T(t)\left(-\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + q\mu^2 I_{Mm} + \sum_{n=1}^{s} b_n\widehat{P}(C^n\otimes\Gamma^n) + \sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\widehat{P}\right) \\
&\alpha(t) + 2\alpha^T(t)\widehat{P}\widehat{H}\beta(t) + \beta^T(t)A_1\beta(t) - \left(\alpha^T(t)\widehat{Z}_1^T + \beta^T(t)\widehat{Z}_2^T\right)F\beta(t) \\
&= \zeta^T(t)\begin{pmatrix} W_1 & E_1 \\ E_1^T & M_2 \end{pmatrix}\zeta(t).
\end{aligned} \tag{30}$$

From (29), we have

$$\eta^T(t)F\beta(t) \ge \mathscr{L}V^1(t) + \beta^T(t)A_1\beta(t). \tag{31}$$

Similarly, we can derive

$$\mathbb{E}\left\{\int_{t_0}^{t_p}\eta^T(t)F\beta(t)\mathrm{d}t\right\} \ge \mathbb{E}\left\{V^1(t_p)\right\} - \mathbb{E}\left\{V^1(t_0)\right\} + \mathbb{E}\left\{\int_{t_0}^{t_p}\beta^T(t)A_1\beta(t)\mathrm{d}t\right\}. \tag{32}$$

Theorem 3. *Network (15) is output-strictly passive if there exist matrices* $F \in \mathbb{R}^{Mq\times Mp}, 0 < A_2 \in \mathbb{R}^{Mq\times Mq}$, *and* $0 < \widehat{P} \in \mathbb{R}^{Mm\times Mm}$ *satisfying*

$$\begin{pmatrix} W_2 & E_2 \\ E_2^T & M_3 \end{pmatrix} \le 0, \tag{33}$$

where $W_2 = W_1 + \widehat{Z}_1^T A_2\widehat{Z}_1, E_2 = E_1 + \widehat{Z}_1^T A_2\widehat{Z}_2$, *and* $M_3 = M_1 + \widehat{Z}_2^T A_2\widehat{Z}_2$.

Proof. We select the same $V^1(t)$ as (19) for network (15), and we can easily obtain

$$\begin{aligned}
&\mathscr{L}V^1(t) - \eta^T(t)F\beta(t) + \eta^T(t)A_2\eta(t) \\
&\le \alpha^T(t)\left(-\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + q\mu^2 I_{Mm} + \sum_{n=1}^{s} b_n\widehat{P}(C^n\otimes\Gamma^n) + \sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\widehat{P}\right) \\
&\alpha(t) + 2\alpha^T(t)\widehat{P}\widehat{H}\beta(t) + \left(\alpha^T(t)\widehat{Z}_1^T + \beta^T(t)\widehat{Z}_2^T\right)A_2\left(\widehat{Z}_1\alpha(t) + \widehat{Z}_2\beta(t)\beta(t)\right) \\
&\quad - \left(\alpha^T(t)\widehat{Z}_1^T + \beta^T(t)\widehat{Z}_2^T\right)F\beta(t) \\
&= \zeta^T(t)\begin{pmatrix} W_2 & E_2 \\ E_2^T & M_3 \end{pmatrix}\zeta(t).
\end{aligned} \tag{34}$$

From (47), one has

$$\eta^T(t)F\beta(t) \ge \mathscr{L}V^1(t) + \eta^T(t)A_2\eta(t). \tag{35}$$

Similarly, we can derive

$$\mathbb{E}\left\{\int_{t_0}^{t_p}\eta^T(t)F\beta(t)\mathrm{d}t\right\}\geq\mathbb{E}\left\{V^1\left(t_p\right)\right\}-\mathbb{E}\left\{V^1\left(t_0\right)\right\}+\mathbb{E}\left\{\int_{t_0}^{t_p}\eta^T(t)A_2\eta(t)\mathrm{d}t\right\}. \tag{36}$$

□

3.3. Synchronization in Passive MWCSNNs

Definition 4. The network (8) with input $\beta_z(t)=0\,(z=1,2,\ldots,M)$ achieves synchronization if

$$\lim_{t\longrightarrow+\infty}\mathbb{E}\left\{\left\|\left(\kappa_z(t)-s(t)\right)\right\|\right\}=0,\quad z=1,2,\ldots,M. \tag{37}$$

Theorem 4. *If network (15) is output-strictly passive with regard to storage function $K(t)=V^1(t)/2$ and $Z_1\in\mathbb{R}^{m\times m}$ is nonsingular, then MWCSNNs (8) can achieve synchronization.*

Proof. If network (15) is output-strictly passive with respect to storage function $K(t)$, then there exist matrices $\mathbb{R}^{Mm\times Mm}\ni A_2>0$ and $F\in\mathbb{R}^{Mm\times Mm}$ such that

$$\mathbb{E}\left\{\int_t^{t+\delta}\eta^T(s)F\beta(s)\mathrm{d}s\right\}\geq\mathbb{E}\{K(t+\delta)\}-\mathbb{E}\{K(t)\}+\mathbb{E}\left\{\int_t^{t+\delta}\eta^T(s)A_2\eta(s)\mathrm{d}s\right\}, \tag{38}$$

for any $t\in\mathbb{R}^+$ and $\delta>0$. Then, we can easily obtain

$$\mathbb{E}\left\{\frac{K(t+\delta)-K(t)}{\varepsilon}\right\}\leq\mathbb{E}\left\{\frac{\int_t^{t+\delta}\eta^T(s)F\beta(s)\mathrm{d}s}{\varepsilon}-\frac{\int_t^{t+\delta}\eta^T(s)A_2\eta(s)\mathrm{d}s}{\varepsilon}\right\}. \tag{39}$$

By taking limit $\delta\longrightarrow0$ in (39), one has

$$\mathbb{E}\left\{\dot{K}(t)\leq\mathbb{E}\eta^T(s)\right\}F\beta(s)-\eta^T(s)A_2\eta(s). \tag{40}$$

Letting $\beta_z(t)=0\,(z=1,2,\ldots,M)$, we can get from (40) that

$$\begin{aligned}\mathbb{E}\left\{\dot{K}(t)\leq\right.&\left.-\mathbb{E}\alpha^T(t)\widehat{Z}_1^TA_2\widehat{Z_1}\alpha(t)\right\}\\&\leq-\lambda_m\left(Z_1^TZ_1\right)\lambda_m(A_2)\mathbb{E}\left\{\|\alpha(t)\|^2\right\}.\end{aligned} \tag{41}$$

From (41) and definition of $V^1(t)$, we can get that $\lim_{t\longrightarrow+\infty}\mathbb{E}\{K(t)\}$ exists. Therefore, we can conclude that

$$\lim_{t\longrightarrow+\infty}\mathbb{E}\left\{\|\alpha(t)\|^2\right\}=\theta\geq0. \tag{42}$$

Suppose that

$$\lim_{t\longrightarrow+\infty}\mathbb{E}\left\{\|\alpha(t)\|^2\right\}=\theta>0. \tag{43}$$

Then, there obviously exists $0<t^*\in\mathbb{R}$ such that

$$\mathbb{E}\left\{\|\alpha(t)\|^2\right\}>\frac{\theta}{2},\quad t\geq t^*. \tag{44}$$

By (41) and (44), we have

$$\mathbb{E}\left\{\dot{K}(t)\right\}<-\frac{\lambda_M\left(Z_1^TZ_1\right)\lambda_M(A_2)\theta}{2},\quad t\geq t^*. \tag{45}$$

From (45), we derive

$$\begin{aligned}\mathbb{E}\{-K(t^*)\}&\leq\mathbb{E}\{K(+\infty)-K(t^*)\}\\&=\int_{t^*}^{+\infty}\mathbb{E}\left\{\dot{K}(t)\right\}\mathrm{d}t\\&<-\int_{t^*}^{+\infty}\frac{\lambda_M\left(Z_1^TZ_1\right)\lambda_M(A_2)\theta}{2}\\&=-\infty,\end{aligned} \tag{46}$$

which results in a contradiction.

Hence, $\lim_{t\longrightarrow+\infty}\mathbb{E}\{\|\alpha(t)\|\}=0$. That is, network (8) realizes synchronization.

The following conclusion can be obtained from Theorems 3 and 4. □

Corollary 1. *Network (8) achieves synchronization if there exist matrices $F\in\mathbb{R}^{Mm\times Mm}$, $0<A_2\in\mathbb{R}^{Mm\times Mm}$, and $0<\widehat{P}\in\mathbb{R}^{Mm\times Mm}$ satisfying*

$$\begin{pmatrix}W_2&E_2\\E_2^T&M_3\end{pmatrix}\leq0, \tag{47}$$

where $W_2=-\widehat{P}\widehat{D}-\widehat{D}\widehat{P}+\widehat{P}\widehat{G}\widehat{G}^T\widehat{P}+\widehat{\xi}+q\mu^2I_{Mm}+\sum_{n=1}^s b_n\widehat{P}(C^n\otimes\Gamma^n)+\sum_{n=1}^s b_n(C^n\otimes\Gamma^n)\widehat{P}+\widehat{Z}_1^TA_2\widehat{Z_1}$, $E_2=\widehat{P}\widehat{H}-(\widehat{Z}_1^TF)/2+\widehat{Z}_1^TA_2\widehat{Z}_2$, $M_3=-(\widehat{Z}_2^TF+F^T\widehat{Z}_2)/2+\widehat{Z}_2^TA_2\widehat{Z}_2$, *and* $q=\lambda_M(\widehat{P})$.

4. Passivity for CSNNMDCs

4.1. Network Model. The CSNNMDCs in this paper is considered as follows:

$$\begin{aligned} d\kappa_z(t) = &\left(-D\kappa_z(t) + Gf(\kappa_z(t)) + B + \sum_{n=1}^{s}\sum_{h=1}^{M} b_n C_{zh}^n \Gamma^n \kappa_h(t-\tau_n) + H\beta_z(t)\right)dt \\ &+ \sigma(\kappa_z(t))d\omega(t), \quad z = 1, 2, \ldots, M, \end{aligned} \tag{48}$$

where $\tau_n (n = 1, 2, \ldots, s)$ are coupling delays and $\kappa_z(t), \beta_z(t), f(\kappa_z(t)), D, G, B, H, b_n, C_{zh}^n$, and Γ^n have the same meanings as those in Section 3.

Suppose that $s(t) = (s_1(t), s_2(t), \ldots, s_m(t))^T \in \mathbb{R}^m$ is an arbitrary desired solution of the isolated node of system (48), then it satisfies

$$ds(t) = (-Ds(t) + Gf(s(t)) + B)dt + \sigma(s(t))d\omega(t). \tag{49}$$

Letting $\alpha_z(t) = (\alpha_{z1}(t), \alpha_{z2}(t), \ldots, \alpha_{zm}(t))^T = \kappa_z(t) - s(t)$, we can obtain from (48) and (49) that

$$\begin{aligned} d\alpha_z(t) = &\left(-D\alpha_z(t) + Gf(\kappa_z(t)) - Gf(s(t)) + \sum_{n=1}^{s}\sum_{j=1}^{M} b_n C_{zj}^n \Gamma^n \alpha_j(t-\tau_n) + H\beta_z(t)\right)dt \\ &+ (\sigma(\kappa_z(t)) - \sigma(s(t)))d\omega(t). \end{aligned} \tag{50}$$

The output vector $\eta_z(t) \in \mathbb{R}^q$ of network (48) is defined as follows:

$$\eta_z(t) = Z_1\alpha_z(t) + Z_2\beta_z(t), \tag{51}$$

where $Z_1 \in \mathbb{R}^{q\times m}$ and $Z_2 \in \mathbb{R}^{q\times p}$.

According to (50), we have

$$\begin{aligned} d\alpha(t) = &\left(-\widehat{D}\alpha(t) + \widehat{G}(f(\kappa(t)) - f(s(t))) + \widehat{H}\beta(t) + \sum_{n=1}^{s} b_n (C^n \otimes \Gamma^n)\alpha(t-\tau_n)\right)dt + \widehat{\sigma}(t)d\widehat{\omega}(t), \\ \eta(t) = &\widehat{Z_1}\alpha(t) + \widehat{Z_2}\beta(t), \end{aligned} \tag{52}$$

where $\widehat{\sigma}(t) = I_M \otimes (\sigma(\kappa(t)) - \sigma(s(t)))$.

Theorem 5. *Network (51) is passive if there exist* $F \in \mathbb{R}^{Mq\times Mp}, 0 < \widehat{P} \in \mathbb{R}^{Mm\times Mm}$, *and* $\widehat{N}_n = diag(N_n^1, N_n^2, \ldots, N_n^M) \in \mathbb{R}^{Mm\times Mm}, n = 1, 2, \ldots, s$, *satisfying*

$$\begin{pmatrix} W_3 & E_3 \\ E_3^T & M_4 \end{pmatrix} \le 0, \tag{53}$$

where $W_3 = -\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + q\mu^2 I_{Mm} + \sum_{n=1}^{s} b_n (\widehat{P}(C^n \otimes \Gamma^n))\widehat{N}_n^{-1}((C^n \otimes \Gamma^n)\widehat{P}) + \sum_{n=1}^{s} b_n \widehat{N}_n, E_3 = \widehat{P}\widehat{H} - (\widehat{Z}_1^T F)/2, M_4 = -(\widehat{Z}_2^T F + F^T \widehat{Z}_2)/2$, *and* $q = \lambda_M(\widehat{P})$.

Proof. For convenience, we denote

$$y(t) = -\widehat{D}\alpha(t) + \widehat{G}(f(\kappa(t)) - f(s(t))) + \widehat{H}\beta(t) + \sum_{n=1}^{s} b_n (C^n \otimes \Gamma^n)\alpha(t-\tau_n), \tag{54}$$

then network (51) can be rewritten as

$$d\alpha(t) = y(t)dt + \widehat{\sigma}(t)d\widehat{\omega}(t). \tag{55}$$

Choose the following Lyapunov functional for network (51):

$$V^2(t) = \alpha^T(t)\widehat{P}\alpha(t) + \sum_{n=1}^{s} b_n \int_{t-\tau_n}^{t} \alpha^T(h)\widehat{N}_n\alpha(h)\mathrm{d}h. \quad (56)$$

In light of Lemma 1, we can obtain

$$\begin{aligned}\mathscr{L}V^2(t) &= V_t^2(t) + V_\alpha^2(t)y(t) + \frac{1}{2}\mathrm{Tr}\left(\widehat{\sigma}^T(t)V_{\alpha\alpha}^2(t)\widehat{\sigma}(t)\right)\\ &= \sum_{n=1}^{s} b_n\alpha^T(t)\widehat{N}_n\alpha(t) + 2\alpha^T(t)\widehat{P}\left(-\widehat{D}\alpha(t) + \widehat{H}\beta(t) + \widehat{G}(f(\kappa(t)) - f(s(t))) + \sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\alpha(t-\tau_n)\right)\\ &\quad - \sum_{n=1}^{s} b_n\alpha^T(t-\tau_n)\widehat{N}_n\alpha(t-\tau_n) + \mathrm{Tr}\left(\widehat{\sigma}^T(t)\widehat{P}\widehat{\sigma}(t)\right).\end{aligned} \quad (57)$$

Moreover,

$$\begin{aligned}2\alpha^T(t)\widehat{P}\left(\sum_{n=1}^{s} b_n(C^n\otimes\Gamma^n)\alpha(t-\tau_n)\right) &\leqslant \sum_{n=1}^{s} b_n\alpha^T(t)\left(\left(\widehat{P}(C^n\otimes\Gamma^n)\right)\widehat{N}_n^{-1}\left((C^n\otimes\Gamma^n)\widehat{P}\right)\right)\alpha(t)\\ &\quad + \sum_{n=1}^{s} b_n\alpha^T(t-\tau_n)\widehat{N}_n\alpha(t-\tau_n).\end{aligned} \quad (58)$$

From (57) and (58), we have

$$\begin{aligned}\mathscr{L}V^2(t) \leq \alpha^T(t)&\left(-\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + \mu^2 qI_{Mm} + \sum_{n=1}^{s} b_n\left(\widehat{P}(C^n\otimes\Gamma^n)\right)\widehat{N}_n^{-1}\left((C^n\otimes\Gamma^n)\widehat{P}\right) + \sum_{n=1}^{s} b_n\widehat{N}_n\right)\\ &\alpha(t) + 2\alpha^T(t)\widehat{P}\widehat{H}\beta(t).\end{aligned} \quad (59)$$

From (59), we have

$$\begin{aligned}&\mathscr{L}V^2(t) - \eta^T(t)F\beta(t) \leq \alpha^T(t)\left(-\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + \mu^2 qI_{Mm} + \sum_{n=1}^{s} b_n\left(\widehat{P}(C^n\otimes\Gamma^n)\right)\widehat{N}_n^{-1}\left((C^n\otimes\Gamma^n)\widehat{P}\right) + \sum_{n=1}^{s} b_n\widehat{N}_n\right)\\ &\alpha(t) + 2\alpha^T(t)\widehat{P}\widehat{H}\beta(t) - \left(\alpha^T(t)\widehat{Z}_1^T + \beta^T(t)\widehat{Z}_2^T\right)F\beta(t)\\ &= \zeta^T(t)\begin{pmatrix} W_3 & E_3\\ E_3^T & M_4\end{pmatrix}\zeta(t).\end{aligned} \quad (60)$$

By (53), we can acquire

$$\mathbb{E}\left\{\int_{t_0}^{t_p} \eta^T(t)F\beta(t)\mathrm{d}t\right\} \geq \mathbb{E}\left\{V^2\left(t_p\right)\right\} - \mathbb{E}\left\{V^2\left(t_0\right)\right\}. \quad (61)$$

By employing similar proof methods in Theorem 5, we can get the following conclusions. □

Theorem 6. *Network (51) is input-strictly passive if there exist matrices* $F \in \mathbb{R}^{Mq\times Mp}, 0 < A_1 \in \mathbb{R}^{Mp\times Mp}, 0 < \widehat{P} \in \mathbb{R}^{Mm\times Mm}$, *and* $\widehat{N}_n = diag(N_n^1, N_n^2, \ldots, N_n^M) \in \mathbb{R}^{Mm\times Mm}$, $n = 1, 2, \ldots, s$, *satisfying*

$$\begin{pmatrix} W_3 & E_3 \\ E_3^T & M_5 \end{pmatrix} \leq 0, \quad (62)$$

where W_3 *and* E_3 *have the same meanings as those in Theorem 5 and* $M_5 = A_1 + M_4$.

Theorem 7. *Network (51) is output-strictly passive if there exist matrices* $F \in \mathbb{R}^{Mq\times Mp}, 0 < A_2 \in \mathbb{R}^{Mq\times Mq}, 0 < \widehat{P} \in \mathbb{R}^{Mm\times Mm}$, *and* $\widehat{N}_n = diag(N_n^1, N_n^2, \ldots, N_n^M) \in \mathbb{R}^{Mm\times Mm}, n = 1, 2, \ldots, s$, *satisfying*

$$\begin{pmatrix} W_4 & E_4 \\ E_4^T & M_6 \end{pmatrix} \leq 0, \quad (63)$$

where $W_4 = W_3 + \widehat{Z}_1^T A_2 \widehat{Z}_1, E_4 = E_3 + \widehat{Z}_1^T A_2 \widehat{Z}_2$, *and* $M_6 = M_4 + \widehat{Z}_2^T A_2 \widehat{Z}_2$.

4.2. Synchronization in Passive CSNNMDCs

Theorem 8. *The CSNNMDCs (48) can achieve synchronization if network (51) is output-strictly passive with regard to storage function* $K(t) = V^2(t)/2$, *and* $Z_1 \in \mathbb{R}^{m\times m}$ *is nonsingular.*

Proof. The results can be easily obtained by employing similar proof method in the Theorem 4.

The following conclusion can be obtained from Theorems 7 and 8. □

Corollary 2. *Network (48) achieves synchronization if there exist matrices* $F \in \mathbb{R}^{Mm\times Mm}, 0 < A_2 \in \mathbb{R}^{Mm\times Mm}$, $0 < \widehat{P} \in \mathbb{R}^{Mm\times Mm}$, *and* $\widehat{N}_n = diag(N_n^1, N_n^2, \ldots, N_n^M) \in \mathbb{R}^{Mm\times Mm}$, $n = 1, 2, \ldots, s$, *satisfying*

$$\begin{pmatrix} W_4 & E_4 \\ E_4^T & M_6 \end{pmatrix} \leq 0, \quad (64)$$

where $W_4 = -\widehat{P}\widehat{D} - \widehat{D}\widehat{P} + \widehat{P}\widehat{G}\widehat{G}^T\widehat{P} + \widehat{\xi} + q\mu^2 I_{Mm} + \sum_{n=1}^s b_n (\widehat{P}(C^n \otimes \Gamma^n))\widehat{N}_n^{-1}((C^n \otimes \Gamma^n)\widehat{P}) + \sum_{n=1}^s b_n \widehat{N}_n + \widehat{Z}_1^T A_2 \widehat{Z}_1, E_4 = \widehat{P}\widehat{H} - (\widehat{Z}_1^T F)/2 + \widehat{Z}_1^T A_2 \widehat{Z}_2, M_6 = -(\widehat{Z}_2^T F + F^T \widehat{Z}_2)/2 + \widehat{Z}_2^T A_2 \widehat{Z}_2$, *and* $q = \lambda_M(\widehat{P})$.

Remark 2. In this paper, two types of network models are proposed (see (8) and (48)), some sufficient conditions for ensuring the passivity of networks (8) and (48) are acquired by employing the stochastic analysis techniques and Lyapunov functional method (see Theorems 1–3 and Theorems 5–7), and several synchronization criteria for networks (8) and (48) are established in view of the output-strictly passivity (see Corollaries 1 and 2).

5. Numerical Examples

Example 1. Take the following MWCSNNs into consideration:

$$\begin{aligned} \mathrm{d}\kappa_z(t) = &\left(-D\kappa_z(t) + Gf(\kappa_z(t)) + B + 0.4\sum_{h=1}^{6} C_{zh}^1 \Gamma^1 \kappa_h(t) + 0.2\sum_{h=1}^{6} C_{zh}^2 \Gamma^2 \kappa_h(t) + 0.1\sum_{h=1}^{6} C_{zh}^3 \Gamma^3 \kappa_h(t) + H\beta_z(t)\right)\mathrm{d}t \\ &+ \sigma(\kappa_z(t))\mathrm{d}\omega(t), \end{aligned} \quad (65)$$

where $z = 1, 2, \ldots, 6$, $f_i(j) = 0.25(|j+1| - |j-1|)$, $i = 1, 2, 3$, $D = \mathrm{diag}(0.5, 0.2, 0.4), B = (0, 0, 0)^T, \Gamma^1 = \mathrm{diag}(0.9, 0.4, 0.3)$, $\Gamma^2 = \mathrm{diag}(0.8, 0.3, 0.4), \Gamma^3 = \mathrm{diag}(0.6, 0.3, 0.8)$, $\sigma(\kappa_z(t)) = \mathrm{diag}(0.2\kappa_{z1}(t), 0.4\kappa_{z2}(t), 0.3\kappa_{z3}(t))$,

$$G = \begin{pmatrix} 0.3 & 0.5 & 0.1 \\ 0.2 & 0.1 & 0.2 \\ 0.12 & 0.1 & 0.4 \end{pmatrix},$$
$$H = \begin{pmatrix} 0.4 & 0.6 \\ 0.8 & 0.3 \\ 0.5 & 0.7 \end{pmatrix},$$
$$C^1 = \begin{pmatrix} 0.8 & 0.2 & 0 & 0.1 & 0.2 & 0.3 \\ 0.2 & -0.6 & 0.1 & 0.2 & 0.1 & 0 \\ 0 & 0.1 & -0.5 & 0 & 0.3 & 0.1 \\ 0.1 & 0.2 & 0 & -0.8 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.3 & 0.2 & -0.9 & 0.1 \\ 0.3 & 0 & 0.1 & 0.3 & 0.1 & -0.8 \end{pmatrix},$$
$$C^2 = \begin{pmatrix} -0.5 & 0.2 & 0 & 0.1 & 0.1 & 0.1 \\ 0.2 & 0.6 & 0.1 & 0.1 & 0.2 & 0 \\ 0 & 0.1 & -0.8 & 0 & 0.3 & 0.4 \\ 0.1 & 0.1 & 0 & -0.6 & 0.1 & 0.3 \\ 0.1 & 0.2 & 0.3 & 0.1 & -0.8 & 0.1 \\ 0.1 & 0 & 0.4 & 0.3 & 0.1 & -0.9 \end{pmatrix},$$
$$C^3 = \begin{pmatrix} -0.9 & 0.4 & 0 & 0.2 & 0.2 & 0.1 \\ 0.4 & -0.8 & 0.2 & 0.1 & 0.1 & 0 \\ 0 & 0.2 & -0.5 & 0 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0 & -0.8 & 0.4 & 0.1 \\ 0.2 & 0.1 & 0.1 & 0.4 & -0.9 & 0.1 \\ 0.1 & 0 & 0.2 & 0.1 & 0.1 & -0.5 \end{pmatrix}. \tag{66}$$

Apparently, $s(t) = (0,0,0)^T \in \mathbb{R}^3$ is a desired solution of an isolated node of network (65), and $f_i(\cdot)$ meets Assumption 1 with $\xi_l = 0.5$.

The $\eta_z(t) \in \mathbb{R}^3$ is selected as

$$\eta_z(t) = Z_1\alpha_z(t) + Z_2\beta_z(t), \quad z = 1,2,\ldots,6, \tag{67}$$

where $\beta_z(t) = (0.6z * \sin(t), 0.4z * \sin(t))^T$,

$$Z_1 = \begin{pmatrix} 0.3 & 0.1 & 0.5 \\ 0.8 & 0.5 & 0.4 \\ 0.4 & 0.2 & 0.7 \end{pmatrix},$$
$$Z_2 = \begin{pmatrix} 0.7 & 0.6 \\ 0.8 & 0.5 \\ 0.2 & 0.3 \end{pmatrix}. \tag{68}$$

Case 1: the following matrices F and $\widehat{P}$ can be obtained:

$$F = I_6 \otimes \begin{pmatrix} -0.4836 & 0.3568 \\ 1.5425 & -0.9371 \\ 0.0897 & 2.7229 \end{pmatrix},$$
$$\widehat{P} = I_6 \otimes \begin{pmatrix} 0.6443 & -0.0057 & -0.0834 \\ -0.0057 & 0.7983 & -0.2136 \\ -0.0834 & -0.2136 & 1.1224 \end{pmatrix}. \tag{69}$$

From Theorem 1, network (65) is passive and Figures 1 and 2 display the simulation results.

Case 2: the following matrices F, $\widehat{P}$, and A_2 that satisfy the condition of Theorem 3 can be obtained:

$$F = I_6 \otimes \begin{pmatrix} 41.6646 & 36.6249 \\ 4.1455 & -8.3129 \\ -30.5560 & -11.0012 \end{pmatrix},$$
$$\widehat{P} = I_6 \otimes \begin{pmatrix} 1.9525 & 0.3124 & -1.6051 \\ 0.3124 & 3.8069 & -3.2750 \\ -1.6051 & -3.2750 & 6.8604 \end{pmatrix}, \tag{70}$$
$$A_2 = I_6 \otimes \begin{pmatrix} 20.0009 & -0.5782 & -12.8691 \\ -0.5782 & 1.0321 & -0.9658 \\ -12.8691 & -0.9658 & 10.1673 \end{pmatrix}.$$

From Theorem 4, network (65) is synchronized, Figure 3 demonstrates the effectiveness and correctness of the obtained results.

Example 2. Take the following CSNNMDCs into consideration:

$$\begin{aligned} d\kappa_z(t) = \Bigg(& -D\kappa_z(t) + Gf(\kappa_z(t)) + B + 0.2\sum_{h=1}^{6} C_{zh}^1\Gamma^1\kappa_h(t-0.5) \\ & + 0.1\sum_{h=1}^{6} C_{zh}^2\Gamma^2\kappa_h(t-0.2) + 0.5\sum_{h=1}^{6} C_{zh}^3\Gamma^3\kappa_h(t-0.3) + H\beta_z(t)\Bigg)dt \\ & + \sigma(\kappa_z(t))d\omega(t), \end{aligned} \tag{71}$$

where $z = 1,2,\ldots,6$, $f_i(j) = 0.25(|j+1| - |j-1|)$, $i = 1,2,3$, $D = \mathrm{diag}(0.8, 0.9, 0.7)$, $B = (0,0,0)^T$, $\Gamma^1 = \mathrm{diag}(0.5, 0.3, 0.2)$, $\Gamma^2 = \mathrm{diag}(0.2, 0.5, 0.3)$, $\Gamma^3 = \mathrm{diag}(0.3, 0.2, 0.5)$, $\sigma(\kappa_z(t)) = \mathrm{diag}(0.4\kappa_{z1}(t), 0.3\kappa_{z2}(t), 0.5\kappa_{z3}(t))$,

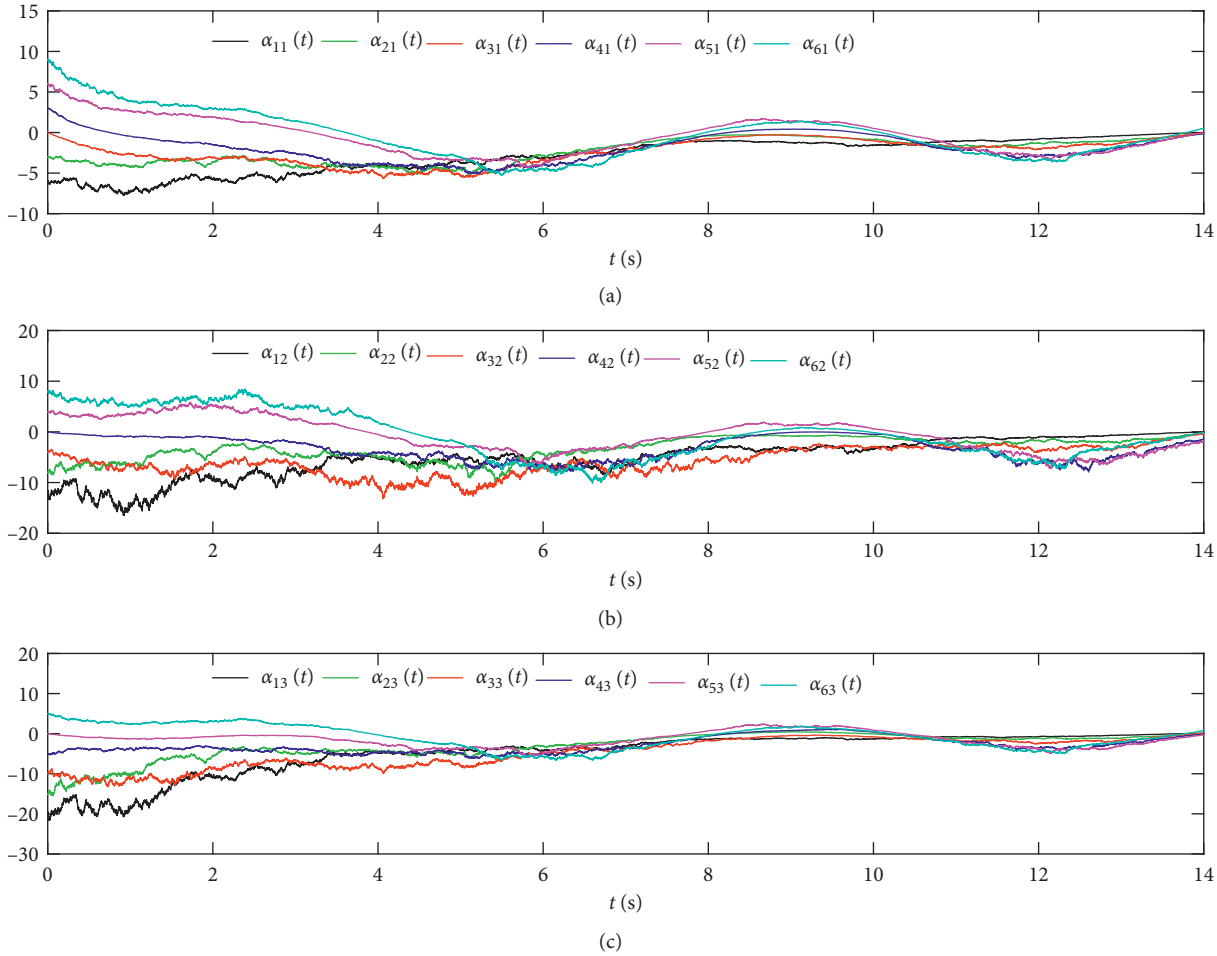


FIGURE 1: $\alpha_{z1}(t), \alpha_{z2}(t)$, and $\alpha_{z3}(t), z = 1, 2, \ldots, 6$.

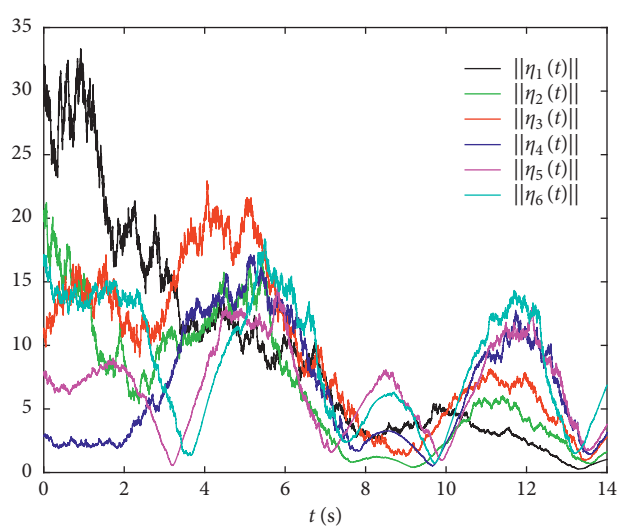


FIGURE 2: $\|\eta_z(t)\|, z = 1, 2, \ldots, 6$.

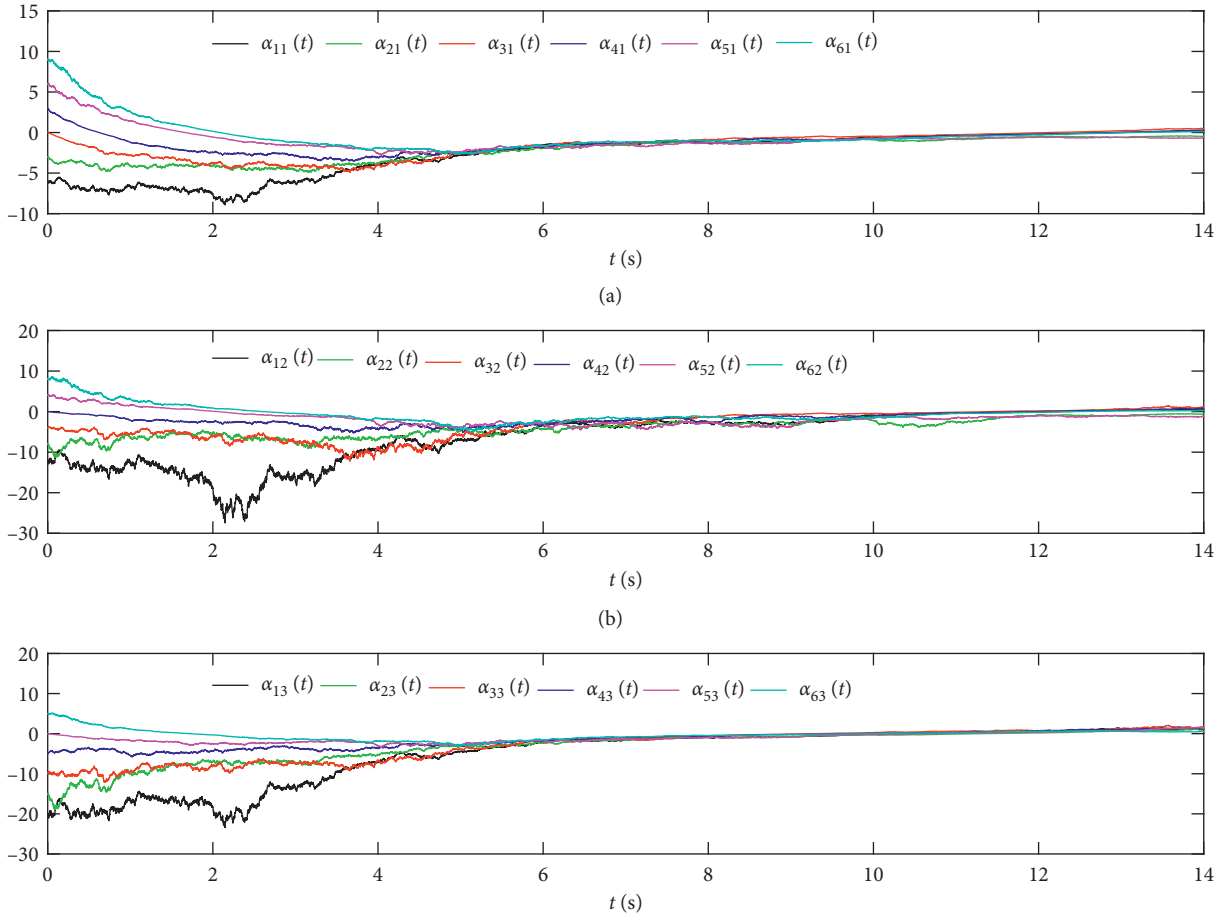


Figure 3: $\alpha_{z1}(t), \alpha_{z2}(t)$, and $\alpha_{z3}(t), z = 1, 2, \ldots, 6$.

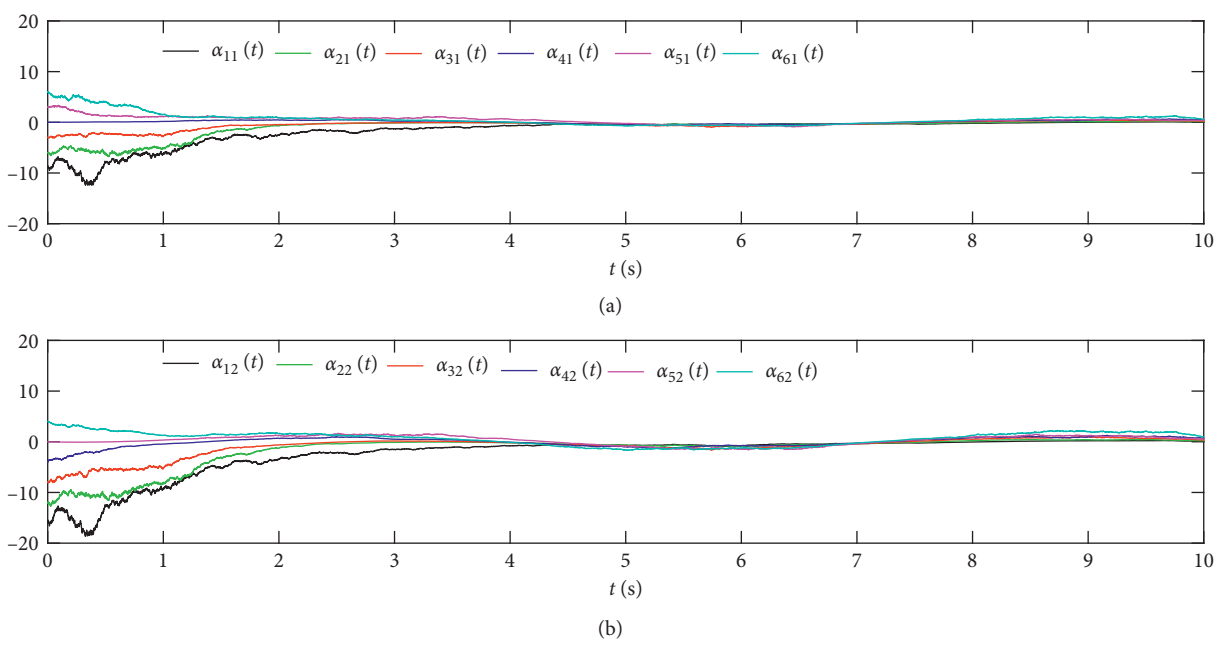
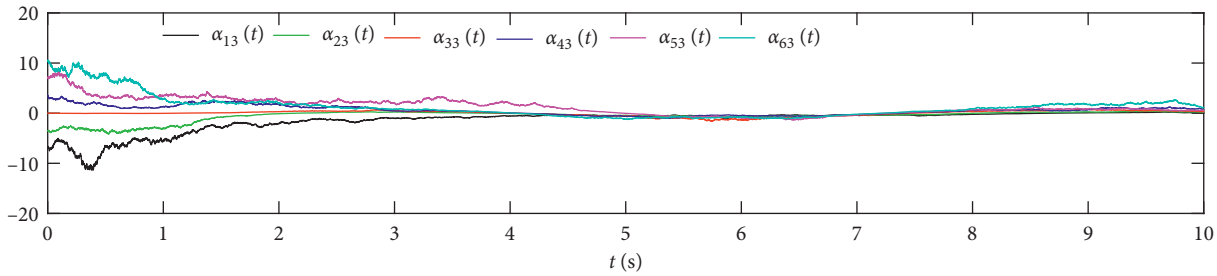


Figure 4: Continued.



(c)

FIGURE 4: $\alpha_{z1}(t), \alpha_{z2}(t)$, and $\alpha_{z3}(t), z = 1, 2, \ldots, 6$.

$$G = \begin{pmatrix} 0.1 & 0.3 & 0.1 \\ 0.2 & 0.1 & 0.3 \\ 0.1 & 0.2 & 0.1 \end{pmatrix},$$

$$H = \begin{pmatrix} 0.2 & 0.1 \\ 0.5 & 0.3 \\ 0.1 & 0.4 \end{pmatrix},$$

$$C^1 = \begin{pmatrix} -0.6 & 0.2 & 0.1 & 0 & 0.2 & 0.1 \\ 0.2 & -0.8 & 0.1 & 0.2 & 0.1 & 0.2 \\ 0.1 & 0.1 & -0.7 & 0 & 0.3 & 0.2 \\ 0 & 0.2 & 0 & -0.5 & 0.2 & 0.1 \\ 0.2 & 0.1 & 0.3 & 0.2 & -0.8 & 0 \\ 0.1 & 0.2 & 0.2 & 0.1 & 0 & -0.6 \end{pmatrix},$$

$$C^2 = \begin{pmatrix} -0.4 & 0.2 & 0 & 0.1 & 0 & 0.1 \\ 0.2 & -0.5 & 0.1 & 0 & 0.2 & 0 \\ 0 & 0.1 & -0.6 & 0 & 0.3 & 0.2 \\ 0.1 & 0 & 0 & -0.5 & 0.1 & 0.3 \\ 0 & 0.2 & 0.3 & 0.1 & -0.7 & 0.1 \\ 0.1 & 0 & 0.2 & 0.3 & 0.1 & -0.7 \end{pmatrix}, \tag{72}$$

$$C^3 = \begin{pmatrix} -0.5 & 0.1 & 0.2 & 0.1 & 0 & 0.1 \\ 0.1 & -0.7 & 0.2 & 0.1 & 0 & 0.3 \\ 0.2 & 0.2 & -0.5 & 0 & 0.1 & 0 \\ 0.1 & 0.1 & 0 & -0.8 & 0.3 & 0.3 \\ 0 & 0 & 0.1 & 0.3 & -0.5 & 0.1 \\ 0.1 & 0.3 & 0 & 0.3 & 0.1 & -0.8 \end{pmatrix}.$$

Apparently, $s(t) = (0, 0, 0)^T \in \mathbb{R}^3$ is a desired solution of an isolated node of network (67), and $f_i(\cdot)$ satisfies Assumption 1 with $\xi_l = 0.5$.

The $\eta_z(t) \in \mathbb{R}^3$ is selected as

$$\eta_z(t) = Z_1 \alpha_z(t) + Z_2 \beta_z(t), \quad z = 1, 2, \ldots, 6, \tag{73}$$

where $\beta_z(t) = (0.4z * \sin(t), 0.5z * \sin(t))^T$,

$$Z_1 = \begin{pmatrix} 0.5 & 0.7 & 0.9 \\ 0.6 & 0.4 & 0.6 \\ 0.2 & 0.5 & 0.1 \end{pmatrix}, \quad Z_2 = \begin{pmatrix} 0.8 & 0.6 \\ 0.5 & 0.9 \\ 0.7 & 0.4 \end{pmatrix}. \tag{74}$$

The matrices $\widehat{N}_1, \widehat{N}_2$, and $\widehat{N}_3$ are chosen as, respectively,

$$\begin{aligned} \widehat{N}_1 &= I_6 \otimes \operatorname{diag}(0.5, 0.6, 0.3), \\ \widehat{N}_2 &= I_6 \otimes \operatorname{diag}(0.3, 0.4, 0.2), \\ \widehat{N}_3 &= I_6 \otimes \operatorname{diag}(0.2, 0.1, 0.4). \end{aligned} \tag{75}$$

Case 1: the following matrices F and $\widehat{P}$ can be obtained:

$$\begin{aligned} F &= I_6 \otimes \begin{pmatrix} 1.7842 & -0.9653 \\ -2.6726 & 3.2707 \\ 3.1171 & -0.5937 \end{pmatrix}, \\ \widehat{P} &= I_6 \otimes \begin{pmatrix} 1.4460 & -0.0495 & -0.0150 \\ -0.0495 & 1.3677 & -0.0831 \\ -0.0150 & -0.0831 & 1.6143 \end{pmatrix}. \end{aligned} \tag{76}$$

From Theorem 5, network (67) is passive, and Figures 4 and 5 display the simulation results.

Case 2: the following matrices $F, \widehat{P}$, and A_2 that satisfy the condition of Theorem 7 can be obtained.

$$\begin{aligned} F &= I_6 \otimes \begin{pmatrix} 1.6654 & -7.2114 \\ -5.4735 & 10.4853 \\ 12.4571 & 2.6646 \end{pmatrix}, \\ \widehat{P} &= I_6 \otimes \begin{pmatrix} 1.8950 & -0.2464 & 0.1908 \\ -0.2464 & 2.3726 & -0.2827 \\ -0.1908 & -0.2827 & 1.5491 \end{pmatrix}, \\ A_2 &= I_6 \otimes \begin{pmatrix} 1.4514 & -1.7789 & -0.2735 \\ -1.7789 & 2.5172 & -0.2573 \\ -0.2735 & -0.2573 & 1.7342 \end{pmatrix}. \end{aligned} \tag{77}$$

From Theorem 8, network (67) is synchronized, and Figure 6 demonstrates the effectiveness and correctness of the obtained results.

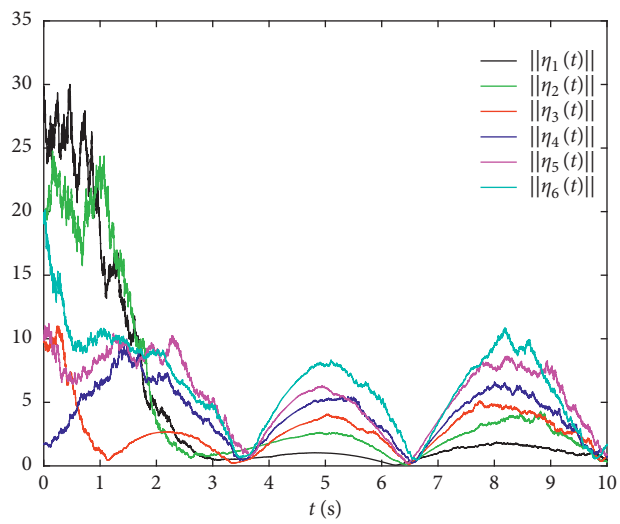


FIGURE 5: $\|\eta_z(t)\|, z = 1, 2, \ldots, 6$.

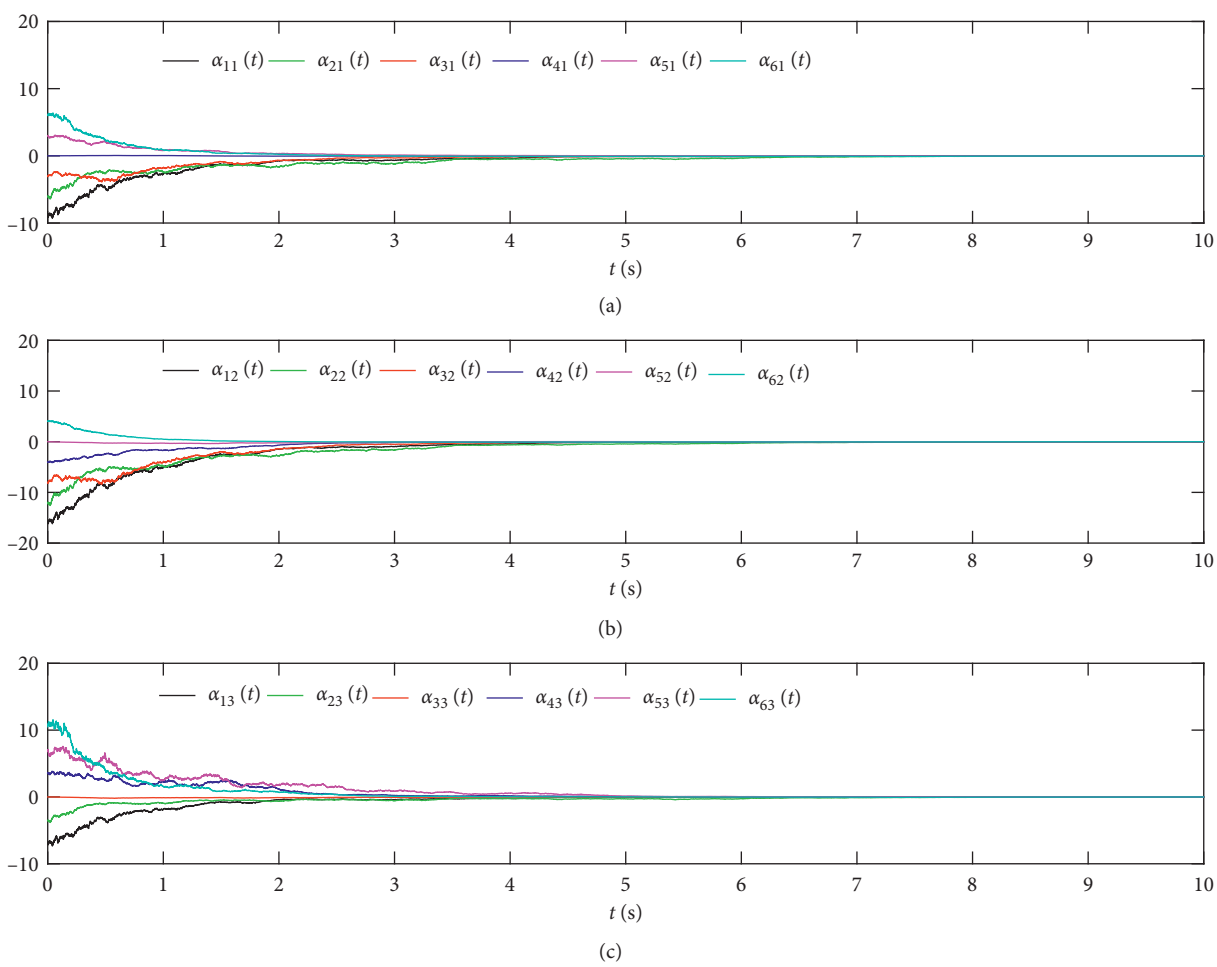


FIGURE 6: $\alpha_{z1}(t), \alpha_{z2}(t)$, and $\alpha_{z3}(t), z = 1, 2, \ldots, 6$.

Remark 3. More recently, some research results on the dynamical behaviors of CSNNs have been obtained, but they all discussed the single weighted network models [41–45]. In this paper, we respectively consider the passivity and synchronization of MWCSNNs (65) and CSNNMDCs (67), which are apparently different from these network models considered in [41–45]. Figures 1 and 2 (Figures 4 and 5) respectively show the change tendencies of $\alpha_{z1}(t), \alpha_{z2}(t), \alpha_{z3}(t), \quad z = 1, 2, \ldots, 6$ and $\|\eta_z(t)\|, \quad z = 1, 2, \ldots, 6$ for networks (65) and (67). From Figures 3 and 6, we can explicitly see that $\alpha_{z1}(t), \alpha_{z2}(t)$, and $\alpha_{z3}(t), \quad z = 1, 2, \ldots, 6$, in networks (65) and (67), respectively, converge to 0 after $12s$ and $5s$, which verify the correctness of the obtained synchronization criteria.

6. Conclusion

Two kinds of MWCSNNs models have been proposed, in which the dimension of output is incompatible with input. On the one hand, we have analyzed the passivity, input-strict passivity, and output-strict passivity for MWCSNNs by employing stochastic analysis techniques. Moreover, two synchronization criteria for MWCSNNs and CSNNMDCs have been derived on the basis of output-strict passivity. Finally, the correctness of the passivity and synchronization criteria has been verified through two numerical examples.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Tianjin, China, under Grant 19JCYBJC18700.

References

[1] S. Wen, Z. Zeng, T. Huang, Q. Meng, and W. Yao, "Lag synchronization of switched neural networks via neural activation function and applications in image encryption," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1493–1502, 2015.

[2] S. Wen, Z. Zeng, T. Huang, and Y. Zhang, "Exponential adaptive lag synchronization of memristive neural networks via fuzzy method and applications in pseudorandom number generators," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1704–1713, 2014.

[3] Y. Yang and J. Cao, "A feedback neural network for solving convex constraint optimization problems," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 340–350, 2008.

[4] S. Mou, H. Gao, J. Lam, and W. Qiang, "A new criterion of delay-dependent asymptotic stability for Hopfield neural networks with time delay," *IEEE Transactions on Neural Networks*, vol. 19, no. 3, pp. 532–535, 2008.

[5] H. Zhang, Z. Wang, and D. Liu, "A comprehensive review of stability analysis of continuous-time recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1229–1262, 2014.

[6] B. Yang, J. Wang, and J. Wang, "Stability analysis of delayed neural networks via a new integral inequality," *Neural Networks*, vol. 88, pp. 49–57, 2017.

[7] C. Chen, L. Li, H. Peng, Y. Yang, L. Mi, and H. Zhao, "A new fixed-time stability theorem and its application to the fixed-time synchronization of neural networks," *Neural Networks*, vol. 123, pp. 412–419, 2020.

[8] G. Rajchakit, P. Chanthorn, M. Niezabitowski, R. Raja, D. Baleanu, and A. Pratap, "Impulsive effects on stability and passivity analysis of memristor-based fractional-order competitive neural networks," *Neurocomputing*, vol. 417, pp. 290–301, 2020.

[9] S. Saravanan, V. Umesha, M. Syed Ali, and S. Padmanabhan, "Exponential passivity for uncertain neural networks with time-varying delays based on weighted integral inequalities," *Neurocomputing*, vol. 314, pp. 429–436, 2018.

[10] C. Ge, J. H. Park, C. Hua, and C. Shi, "Robust passivity analysis for uncertain neural networks with discrete and distributed time-varying delays," *Neurocomputing*, vol. 364, pp. 330–337, 2019.

[11] Q. Xiao, Z. Huang, and Z. Zeng, "Passivity analysis for memristor-based inertial neural networks with discrete and distributed delays," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 375–385, 2019.

[12] Y. Wang, Y. Cao, Z. Guo, and S. Wen, "Passivity and passification of memristive recurrent neural networks with multi-proportional delays and impulse," *Applied Mathematics and Computation*, vol. 369, p. 124838, 2020.

[13] G. Rajchakit and R. Sriraman, "Robust passivity and stability analysis of uncertain complex-valued impulsive neural networks with time-varying delays," *Neural Processing Letters*, vol. 53, no. 1, pp. 581–606, 2021.

[14] P. Chanthorn, G. Rajchakit, S. Ramalingam, C. P. Lim, and R. Ramachandran, "Robust dissipativity analysis of Hopfield-type complex-valued neural networks with time-varying delays and linear fractional uncertainties," *Mathematics*, vol. 8, no. 4, p. 595, 2020.

[15] J. Zhou, Y. Liu, J. Xia, Z. Wang, and S. Arik, "Resilient fault-tolerant anti-synchronization for stochastic delayed reaction-diffusion neural networks with semi-Markov jump parameters," *Neural Networks*, vol. 125, pp. 194–204, 2020.

[16] Y. Liu, Z. Xuan, Z. Wang, J. Zhou, and Y. Liu, "Sampled-data exponential synchronization of time-delay neural networks subject to random controller gain perturbations," *Applied Mathematics and Computation*, vol. 385, p. 125429, 2020.

[17] Q. Zhu and J. Cao, "Robust exponential stability of Markovian jump impulsive stochastic Cohen-Grossberg neural networks with mixed time delays," *IEEE Transactions on Neural Networks*, vol. 21, no. 8, pp. 1314–1325, 2010.

[18] W. Zhang, Y. Tang, W. K. Wong, and Q. Miao, "Stochastic stability of delayed neural networks with local impulsive effects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2336–2345, 2015.

[19] R. Sriraman, Y. Cao, and R. Samidurai, "Global asymptotic stability of stochastic complex-valued neural networks with probabilistic time-varying delays," *Mathematics and Computers in Simulation*, vol. 171, pp. 103–118, 2020.

[20] D. Yang and X. Li, "Robust stability analysis of stochastic switched neural networks with parameter uncertainties via

state-dependent switching law," *Neurocomputing*, vol. 452, pp. 813–819. in press, 2021.

[21] P. Chanthorn, G. Rajchakit, U. Humphries, P. Kaewmesri, R. Sriraman, and C. P. Lim, "A delay-dividing approach to robust stability of uncertain stochastic complex-valued Hopfield delayed neural networks," *Symmetry*, vol. 12, no. 5, p. 683, 2020.

[22] Z.-G. Wu, P. Shi, H. Su, and J. Chu, "Passivity analysis for discrete-time stochastic Markovian jump neural networks with mixed time delays," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1566–1575, 2011.

[23] R. Samidurai and R. Manivannan, "Robust passivity analysis for stochastic impulsive neural networks with leakage and additive time-varying delay components," *Applied Mathematics and Computation*, vol. 268, pp. 743–762, 2015.

[24] R. Samidurai and R. Manivannan, "Delay-range-dependent passivity analysis for uncertain stochastic neural networks with discrete and distributed time-varying delays," *Neurocomputing*, vol. 185, pp. 191–201, 2016.

[25] G. Nagamani, T. Radhika, and Q. Zhu, "An improved result on dissipativity and passivity analysis of Markovian jump stochastic neural networks with two delay components," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 12, pp. 3018–3031, 2017.

[26] G. Rajchakit, R. Sriraman, and R. Samidurai, "Dissipativity analysis of delayed stochastic generalized neural networks with Markovian jump parameters," *International Journal of Nonlinear Sciences and Numerical Simulation*, 2021, in press.

[27] U. Humphries, G. Rajchakit, R. Sriraman et al., "An extended analysis on robust dissipativity of uncertain stochastic generalized neural networks with Markovian jumping parameters," *Symmetry*, vol. 12, no. 6, pp. 1–21, 2020.

[28] C. M. Gray, "Synchronous oscillations in neuronal systems: mechanisms and functions," *Journal of Computational Neuroscience*, vol. 1, no. 1-2, pp. 11–38, 1994.

[29] J. Feng, S. Wang, and Z. Wang, "Stochastic synchronization in an array of neural networks with hybrid nonlinear coupling," *Neurocomputing*, vol. 74, no. 18, pp. 3808–3815, 2011.

[30] M. Syed Ali, R. Saravanakumar, C. K. Ahn, and H. R. Karimi, "Stochastic $H\infty$ filtering for neural networks with leakage delay and mixed time-varying delays," *Information Sciences*, vol. 388-389, pp. 118–134, 2017.

[31] N. Li and J. Cao, "Passivity and robust synchronisation of switched interval coupled neural networks with time delay," *International Journal of Systems Science*, vol. 47, no. 12, pp. 2827–2836, 2015.

[32] C. Huang, W. Wang, J. Cao, and J. Lu, "Synchronization-based passivity of partially coupled neural networks with event-triggered communication," *Neurocomputing*, vol. 319, pp. 134–143, 2018.

[33] W. Chen, Y. Huang, and S. Ren, "Passivity of coupled memristive delayed neural networks with fixed and adaptive coupling weights," *Neurocomputing*, vol. 313, pp. 346–363, 2018.

[34] S.-Y. Ren, J.-L. Wang, and J. Wu, "Generalized passivity of coupled neural networks with directed and undirected topologies," *Neurocomputing*, vol. 314, pp. 371–385, 2018.

[35] Y. Huang, S. Lin, and E. Yang, "Event-triggered passivity of multi-weighted coupled delayed reaction-diffusion memristive neural networks with fixed and switching topologies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 89, p. 105292, 2020.

[36] W. Wu and T. Chen, "Global synchronization criteria of linearly coupled neural network systems with time-varying coupling," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 319–332, 2008.

[37] X. Qi, H. Bao, and J. Cao, "Synchronization criteria for quaternion-valued coupled neural networks with impulses," *Neural Networks*, vol. 128, pp. 150–157, 2020.

[38] Y. Wang, J. Lu, X. Li, and J. Liang, "Synchronization of coupled neural networks under mixed impulsive effects: a novel delay inequality approach," *Neural Networks*, vol. 127, pp. 38–46, 2020.

[39] S. Chen, H. Jiang, B. Lu, and Z. Yu, "Exponential synchronization for inertial coupled neural networks under directed topology via pinning impulsive control," *Journal of the Franklin Institute*, vol. 357, no. 3, pp. 1671–1689, 2020.

[40] Y.-L. Huang, S.-H. Qiu, and S.-Y. Ren, "Finite-time synchronisation and passivity of coupled memristive neural networks," *International Journal of Control*, vol. 93, no. 12, pp. 2824–2837. in press, 2020.

[41] X. Yang and J. Cao, "Stochastic synchronization of coupled neural networks with intermittent control," *Physics Letters A*, vol. 373, no. 36, pp. 3259–3272, 2009.

[42] M. J. Park, O. M. Kwon, J. H. Park, S. M. Lee, and E. J. Cha, "Synchronization criteria for coupled stochastic neural networks with time-varying delays and leakage delay," *Journal of the Franklin Institute*, vol. 349, no. 5, pp. 1699–1720, 2012.

[43] H. Bao, J. H. Park, and J. Cao, "Exponential synchronization of coupled stochastic memristor-based neural networks with time-varying probabilistic delay coupling and impulsive delay," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 1, pp. 190–201, 2016.

[44] H. Chen, P. Shi, and C.-C. Lim, "Exponential synchronization for Markovian stochastic coupled neural networks of neutral-type via adaptive feedback control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1618–1632, 2017.

[45] H. Li, J.-A. Fang, X. Li, L. Rutkowski, and T. Huang, "Event-triggered impulsive synchronization of discrete-time coupled neural networks with stochastic perturbations and multiple delays," *Neural Networks*, vol. 132, pp. 447–460, 2020.

[46] H.-A. Tang, J.-L. Wang, L. Wang, X. Hu, Y. Zhou, and S. Duan, "Impulsive control for passivity and exponential synchronization of coupled neural networks with multiple weights," *Journal of the Franklin Institute*, vol. 356, no. 10, pp. 5434–5463, 2019.

[47] Y. Wang, Y. Huang, and E. Yang, "Event-triggered communication for passivity and synchronisation of multi-weighted coupled neural networks with and without parameter uncertainties," *IET Control Theory & Applications*, vol. 14, no. 9, pp. 1228–1239, 2020.

[48] J.-L. Wang and L.-H. Zhao, "PD and PI control for passivity and synchronization of coupled neural networks with multi-weights," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 790–802, 2021.

[49] S. Chen, J. Feng, J. Wang, and Y. Zhao, "Almost sure exponential synchronization of drive-response stochastic memristive neural networks," *Applied Mathematics and Computation*, vol. 383, p. 125360, 2020.

[50] L. Huang and X. Mao, "SMC design for robust H_∞ control of uncertain stochastic delay systems," *Automatica*, vol. 46, pp. 405–412, 2010.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 4472174, 15 pages
https://doi.org/10.1155/2021/4472174

Research Article

An Implicit Memory-Based Method for Supervised Pattern Recognition

Yu Ma,[1,2,3] **Shafei Wang**,[1,4] **Junan Yang**,[5] **Yanfei Bao**,[1] **and Jian Yang**[1]

[1]*Academy of Military Science of the People's Liberation Army, Beijing 100000, China*
[2]*Naval Aviation University, Yantai 264000, China*
[3]*Peng Cheng Laboratory, Shenzhen 518000, China*
[4]*School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[5]*Institution of Electronic Countermeasure, National University of Defense Technology, Hefei 230037, China*

Correspondence should be addressed to Shafei Wang; wangshafei1964@126.com

Received 8 May 2021; Revised 24 June 2021; Accepted 9 July 2021; Published 17 July 2021

Academic Editor: Zi-Peng Wang

How the human brain does recognition is still an open question. No physical or biological experiment can fully reveal this process. Psychological evidence is more about describing phenomena and laws than explaining the physiological processes behind them. The need for interpretability is well recognized. This paper proposes a new method for supervised pattern recognition based on the working pattern of implicit memory. The artificial neural network (ANN) is trained to simulate implicit memory. When an input vector is not in the training set, the ANN can treat the input as a "*do not care*" term. The ANN may output any value when the input is a "*do not care*" term since the training process needs to use as few neurons as possible. The trained ANN can be expressed as a function to design a pattern recognition algorithm. Using the Mixed National Institute of Standards and Technology database, the experiments show the efficiency of the pattern recognition method.

1. Introduction

Pattern recognition methods can be divided into two categories: two-stage and end-to-end. Most traditional pattern recognition methods are two-stage: feature extraction and pattern classification [1]. Feature extraction reduces the number of resources that are required to describe a large amount of raw data. The first step is to identify the measurable quantities that make these training sets $\mathbf{X}_1, \ldots, \mathbf{X}_l$ distinct from each other. The measurements used for the classification, such as mean value and the standard deviation, are known as features. In general, some features constitute a feature vector. Some information gets lost since feature extraction is not a lossless compression approach. The lost information cannot be used for pattern recognition. Therefore, how to generate features is a fundamental issue. In the feature vector selection, two crucial issues are the best number of features and the classifier design [2]. For example, feature selection plays an important role in text classification [3]. The complexity of the practice data makes it arduous to use two-stage methods.

Nowadays, deep neural networks can be trained end-to-end [4]. Raw data can reserve all information of the pattern. Inspired by the biological neural networks that constitute animal brains, the artificial neural network (ANN) is applied to do image classification [5], speech separation [6], forest fire prediction [7], etc. However, a major drawback of neural networks is the black-box character. Explaining why the neural networks make a particular decision is formidable. The knowledge representation of neural network is unreadable to humans [8]. The exact reason why trained deep neural networks can implement recognition remains an open question [9]. The training algorithm does not specify the way to recognize. A causal model is formidable to build or acquire. End-to-end learning relies on data for the cognitive task [10, 11], where the data cannot tell the reasons [12].

In a supervised pattern recognition task, a set of training data (training set) is used to train a learning procedure. A training set is a set of instances that are properly labeled by hand with the corrected labels. The learning procedure attempts to recognize the instances as accurately as possible. The goal of the learning procedure is to minimize the error rate on a test set. The question arising in the recognition task is why a new data instance can be classified as a particular category. The problem of supervised pattern recognition can be stated as follows. Assume that the training set $\mathbf{X}_i$ only contains instances with label i where $i \in \mathscr{L} \triangleq \{1, \ldots, l\}$, and $\mathbf{X}_i \cap \mathbf{X}_j = \varnothing$ for any $i \neq j$. Given l training sets $\mathbf{X}_1, \ldots, \mathbf{X}_l$, the question is how to label a new instance $\tilde{\mathbf{x}}$.

A memory system is involved in the process of recognition. Jacoby and Kelly posit that memory can serve as both storage and a tool [13]. Memory is treated as storage in a recall. In this case, the focus is on the past, and memory is used as computer storage. Meanwhile, memory (from experience) can be used as a tool to perceive and interpret present events.

The implicit memory is acquired and used unconsciously and can serve as a tool/function [14–16]. In one experiment, two groups of people are asked several times to solve a Tower of Hanoi puzzle. One group is amnesic patients with heavily impaired long-term memory, and the other is composed of healthy subjects. The first group shows the same improvements over time as the second one, even if some participants claim that they do not even remember seeing the puzzle before. These findings strongly suggest that procedural memory is completely independent of declarative memory [17]. Given a game state, implicit memory is trained to output an operation. Memorizing the solving steps is not necessary. When the state appears again, the input can evoke the trained operator [18].

Usually, humans implement recognition processes unconsciously, and the focus is on current given images. So the memory works as a tool while implementing the recognition processes. The recognition process depends on the similarity comparison between the current input and the labeled instances. The more similar, the more likely they are in the same class and have the same label. However, the way to compare similarities without memorizing any labeled instances is not evident.

This paper proposes an implicit memory-based method for supervised pattern recognition. The method does not memorize or recall any labeled instances and is not in the two-stage or end-to-end categories. The proposed method has interpretability since similarity criteria are used in the process of recognition. A new instance is recognized as a particular class because the instance appears similar enough to the training data of the class. Compared with the k-nearest neighbors algorithm [19], the proposed method does not need to recall and iterate through the training sets. The process is consistent with the human ability of pattern recognition. People may forget most of the training instances, but they can recognize a new instance. The Mixed National Institute of Standards and Technology (MNIST) database (General site for the MNIST database: http://yann.lecun.com/exdb/mnist) is used to verify the proposed method.

The rest of this paper is organized as follows. First, a model is built to describe implicit memory. Second, with the implicit memory model, a recognition algorithm is proposed. Then an application and the analysis of experimental results are given.

Notations: $|\mathbf{X}|$ expresses the cardinality of the set $\mathbf{X}$; $\{0,1\}^n$ is an n-dimensional space constructed by 0 and 1; d is a metric on $\{0,1\}^n$; define a distance function between any point $\mathbf{z}$ of $\{0,1\}^n$ and any nonempty set $\mathbf{X}$ of $\{0,1\}^n$ by

$$d(\mathbf{z}, \mathbf{X}) = \inf\{d(\mathbf{z}, \mathbf{x}) | \mathbf{x} \in \mathbf{X}\}. \tag{1}$$

Given two n-dimensional $\mathbf{a}$ and $\mathbf{b}$, the element-wise product of $\mathbf{a}$ and $\mathbf{b}$, written $\mathbf{a}^{\circ}\mathbf{b}$, is the vector $\mathbf{c}$ with elements given by $c_i = a_i b_i$; $|\mathbf{x}|_{l_1}$ expresses the number of 1 in the binary vector $\mathbf{x}$; the probability of $X = x$ is written as $\Pr\{X = x\}$; $\mathscr{L} \triangleq \{1, \ldots, l\}$; $\mathscr{N} \triangleq \{1, \ldots, n\}$; $\mathscr{P} \triangleq \{1, \ldots, p\}$.

The expression of an inverter (NOT circuit) can be expressed as follows: If the input variable is called Z and the output variable is called Y, then $Y = \overline{Z}$. If $Z = 0$, then $Y = 1$ and if $Z = 1$, then $Y = 0$. The expression of a 2-input AND gate can be expressed in equation form as follows: $Y = Z_1 Z_2$. The output Y of an AND gate is 1 only when both inputs are 1 s. The expression of a 2-input OR gate can be expressed as follows: $Y = Z_1 + Z_2$. The output Y of an OR gate is 1 when any one or more of the inputs are 1 s.

2. Model of Implicit Memory

In this section, a model is built to describe implicit memory. From one given input, implicit memory can give an output. Both the input and output of implicit memory are actual signals. The signals have explicit physical meaning in the real world, such as sound, light, electricity, etc. The types of input-output signals can be the same or different. Meanwhile, the input-output signals can be measured and represented as a binary vector.

The action of implicit memory can be represented as a function ψ: $\mathbf{Z} \longrightarrow \mathbf{Y}$. The domain $\mathbf{Z}$ is a set of binary vectors. The element of $\mathbf{Z}$ represents the input signal of implicit memory. The codomain $\mathbf{Y}$ is also a set of binary vectors. The element of $\mathbf{Y}$ represents the output signal of implicit memory. The set of all ordered pairs $(\mathbf{z}, \mathbf{y})$ represents the training result of implicit memory where $\mathbf{z}$ is an element of $\mathbf{Z}$ and $\mathbf{y}$ is an element of $\mathbf{Y}$. An example of the function is given as follows: the domain is chosen to be the set $\{000, 001, 100, 110\}$, and the codomain is the set $\{01, 10\}$. Figure 1 shows these mapping relationships.

A computer can store all ordered pairs $(\mathbf{z}, \mathbf{y})$ in a database. Given an input, the computer can search the database for output. With the database, the computer can simulate the external behaviors of implicit memory. However, this method needs to recall the ordered pairs stored in the database. This internal process is different from the process of implicit memory. The implicit memory does not need to execute memorizing or searching. The implicit memory is similar in operation to a high-speed logic circuit. From one given input, the implicit memory would not have a second's hesitation of giving output.

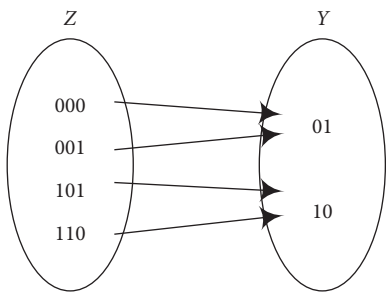


Figure 1: Diagram represents a function with domain $\{000, 001, 101, 110\}$, codomain $\{01, 10\}$, and set of ordered pairs $\{(000, 01), (001, 01), (101, 10), (110, 10)\}$.

A logic circuit and a database have different implementations. In the preparation phase, the database stores the input-output pairs on a hard disk. However, the logic circuit connects the logic gates to implement the input-output maps. At run time, the database searches the given input on the hard disk to find an output. In the logic circuit, nevertheless, the input goes through the logic gates to get the output. The complexity of the practice data also makes it arduous to manually design logic circuits. However, ANN can simulate the actions of implicit memory automatically.

The capacities of the implicit memory can be represented as a set $\mathbf{X} = \{(\mathbf{z}_1, \mathbf{y}_1), \ldots, (\mathbf{z}_n, \mathbf{y}_n)\}$ where $\mathbf{z}_i$ and $\mathbf{y}_i$ are binary vectors. Without loss of generality, assume that all of $\mathbf{z}_1, \ldots, \mathbf{z}_n$ have the same length and all of $\mathbf{y}_1, \ldots, \mathbf{y}_n$ have the same length. And then, the operation of an implicit memory can be expressed with a table. The table lists all allowed input vectors with the corresponding output, as illustrated in Table 1 for an example. The table shows the output for each allowed input. Only the first bit's implementation of the output signal is considered since the other implementations can operate in the same way. The table can serve as a *truth table*.

Truth tables are widely used to describe the operation of logic circuits. With a truth table, the sum-of-products (SOP) expression can be written. The Boolean SOP expression can be obtained from the truth table by ORing the product terms, for which $Y_1 = 1$ is

$$Y_1 = Z_1\overline{Z}_2Z_3 + Z_1Z_2\overline{Z}_3. \tag{2}$$

The first term in the expression is formed by ANDing the three variables Z_1, $\overline{Z}_2$, and Z_3. The second term is formed by ANDing the three variables Z_1, Z_2, and $\overline{Z}_3$. The logic gates can be used to implement the expression. The logic gates are constructed as follows: two inverters to form the $\overline{Z}_2$ and $\overline{Z}_3$ variables; two 3-input AND gates to form the terms $Z_1\overline{Z}_2Z_3$ and $Z_1Z_2\overline{Z}_3$; and one 2-input OR gate to form the final output function, $Z_1\overline{Z}_2Z_3 + Z_1Z_2\overline{Z}_3$. Figure 2 shows the logic diagram. Therefore, a logic circuit can simulate the actions of the implicit memory.

2.1. ANN-Based Boolean Operation. NAND gate is a universal gate because it can be used to produce the NOT, the AND, the OR, and the NOR functions [20]. With NAND gates and appropriate connections, all logic circuits can be built.

Table 1: Example of listing the operation of an implicit memory.

Input			Output	Product term
Z_1	Z_2	Z_3	Y_1	
0	0	0	0	
0	0	1	0	
1	0	1	1	$Z_1\overline{Z}_2Z_3$
1	1	0	1	$Z_1Z_2\overline{Z}_3$

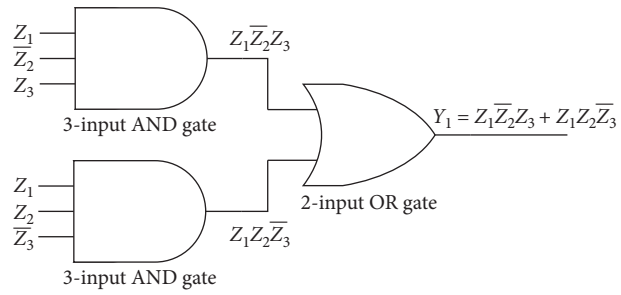


Figure 2: Logic diagram represents $Y_1 = Z_1\overline{Z}_2Z_3 + Z_1Z_2\overline{Z}_3$.

Suppose there is a sigmoid neuron with two inputs, x_1 and x_2. The sigmoid neuron has weights for each input, w_1, w_2, and an overall bias, b. The output of the sigmoid neuron is $\sigma(w_1x_1 + w_2x_2 + b)$, where σ is called the sigmoid function and is defined by

$$\sigma(y) = \frac{1}{1 + e^{y}}. \tag{3}$$

When $w_1 = w_2 = -20$ and $b = 30$, the sigmoid neuron is shown in Figure 3. Then the input 00 produces output 1 since $\sigma(-20 * 0 - 20 * 0 + 30) = \sigma(30) \approx 1$. Similar calculations show that the inputs 01 and 10 produce output 1. But input 11 produces output 0 since $\sigma(-20 * 1 - 20 * 1 + 30) = \sigma(-10) \approx 0$. Therefore, the sigmoid neuron can implement a 2-input NAND gate.

Let the value of w_2 be 0. The output is not affected by the input x_2. The sigmoid neuron equals a 1-input neuron. When $w_1 = -20$ and $b = 10$, the sigmoid neuron is shown in Figure 4. Then input 0 produces output 1 since $\sigma(-20 * 0 + 10) = \sigma(10) \approx 1$. But input 1 produces output 0 since $\sigma(-20 * 1 + 10) = \sigma(-10) \approx 0$. The sigmoid neuron can implement a 1-input NOT gate.

When $w_1 = 20$ and $b = -10$, the sigmoid neuron is shown in Figure 5. Then input 0 produces output 0 since $\sigma(20 * 0 - 10) = \sigma(-10) \approx 0$. But input 1 produces output 1 since $\sigma(20 * 1 - 10) = \sigma(10) \approx 1$. The sigmoid neuron can implement a connecting line.

2.2. Simulation of Implicit Memory. According to De Morgan's laws, Boolean expression can be resolved into 2-input NAND and 1-input NOT. For example,

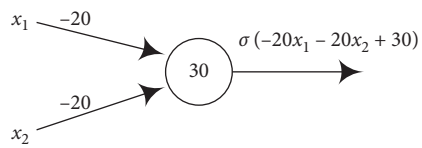


Figure 3: Sigmoid neuron with $w_1 = w_2 = -20$ and $b = 30$.

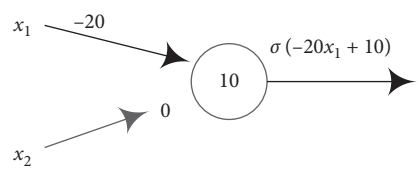


Figure 4: Sigmoid neuron with one input, with weight −20, and an overall bias of 10.

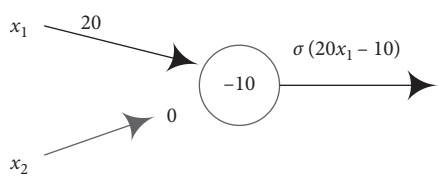


Figure 5: Sigmoid neuron with one input, with weight 20, and an overall bias of −10.

$$\begin{aligned} Y_1 &= Z_1\overline{Z}_2Z_3 + Z_1Z_2\overline{Z}_3 \\ &= \overline{\overline{Z_1\overline{Z}_2Z_3}} + \overline{\overline{Z_1Z_2\overline{Z}_3}} \\ &= \overline{\overline{Z_1\overline{Z}_2Z_3}}\;\overline{\overline{Z_1Z_2\overline{Z}_3}}. \end{aligned} \tag{4}$$

A neural network of five layers can implement expression (4), as is shown in Figure 6. This method of constructing the network has generality. ANN can be configured to execute an arbitrary map. Therefore, ANN can simulate the actions of the implicit memory.

Sometimes a situation arises in which some input variable combinations are not allowed. Because these unallowed states never occur in an application, they can be treated as "*do not care*" terms. For these "*do not care*" terms, either a 1 or a 0 may be assigned to the output. The "*do not care*" terms can be used to simplify an expression. Table 2 shows that, for each "*do not care*" term, a χ is placed in the output. As indicated in Figure 7, when grouping the 1s on the Karnaugh map, the χs can be treated as 1 s to make a larger grouping or as 0 s if they cannot have the advantage. The larger the group, the simpler the resulting term [20]. With "do not care", the neural network of five layers can be simplified to a connection line between Y_1 and Z_1.

Using the ordered pair set $\mathbf{X}$ as a training set, the ANN can be trained to simulate the implicit memory. The optimization process of training the ANN has some similar properties to a Boolean expression simplification process. Their purpose is to implement the required functionality with minimal resources, such as logic gates and activated neurons. The brain might have less than 1% neurons active at any given time [21–23]. If an input vector is not included in the training set, then the ANN can treat the input as a "*do not care*" term. The ANN can output either a 1 or a 0 when the input is a "*do not care*" term.

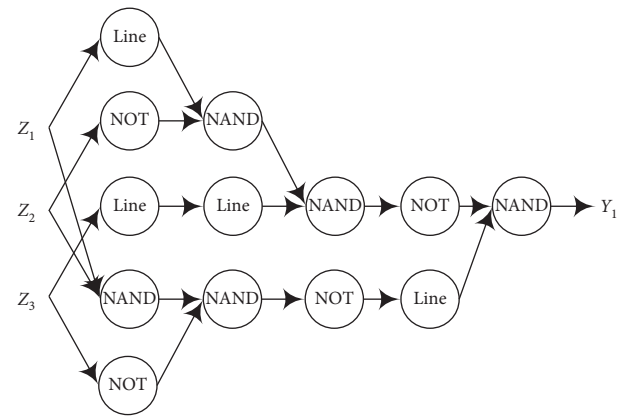


Figure 6: ANN can implement the Boolean expression $Z_1\overline{Z}_2Z_3 + Z_1Z_2\overline{Z}_3$. The neurons are configured to implement the NOT, the NAND, and the connecting line. In general, the neurons have more inputs. While the weight of inputs is 0, the outputs are not affected, which are not drawn.

Table 2: Example of a truth table.

Input			Output	
Z_1	Z_2	Z_3	Y_1	
0	0	0	0	
0	0	1	0	
1	0	1	1	
1	1	0	1	
0	1	0	χ	Do not care
0	1	1	χ	Do not care
1	0	0	χ	Do not care
1	1	1	χ	Do not care

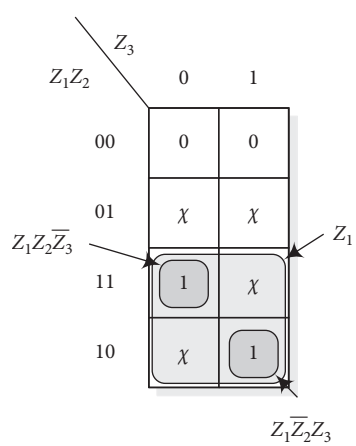


Figure 7: Example of the use of "do not care" conditions to simplify an expression. Without "do not care" $Y_1 = Z_1\overline{Z}_2Z_3 + Z_1Z_2\overline{Z}_3$. With "do not care" $Y_1 = Z_1$. So you can see the advantage of using "do not care" terms to get the simplest expression.

Suppose the supervised learning algorithm trains an ANN. While the given input is in the training set, the output of the trained ANN can be expected. Otherwise, the output of the trained ANN cannot be expected. Only by actual measuring can the output be known. The measurement process is like sampling from a statistical population. The trained ANN model can be expressed as a function f. The function f has the following properties:

(i) The output of the function is a specified value when the input is in the training set. That is, if $(\mathbf{z}, \mathbf{y})$ is in the training set, then $f(\mathbf{z}) = \mathbf{y}$.

(ii) Otherwise, the output can be assigned by generating a sample from a statistical population.

The following section proposes a pattern recognition algorithm with the above function f.

3. Recognition Based on the Implicit Memory Model

The principle of recognition is based on the intuitive assumption that examples in the same class are closer/similar [24, 25]. Therefore, a recognition algorithm is proposed to estimate the similarity via the implicit memory model.

Let's denote a signal by $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, where $x_i \in \{0, 1\}$, $i \in \mathcal{N} \triangleq \{1, \ldots, n\}$. When a function f can precisely predict any masked part of the signal $\mathbf{x}$ and $\tilde{\mathbf{x}}$ is not the same as $\mathbf{x}$, distinguishing between the signal $\mathbf{x}$ and any other signal $\tilde{\mathbf{x}}$ is feasible. The following theorem describes how to recognize the signal $\mathbf{x}$.

Theorem 1. *Consider the signal $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is a constant. For an arbitrary mask $\mathbf{m} = [m_1, \ldots, m_n] \in \{0, 1\}^n$, a function g satisfies $g(\mathbf{m}) = \mathbf{x}^\circ(1 - \mathbf{m})$ where $(1 - \mathbf{m}) = [1 - m_1, \ldots, 1 - m_n]$. Construct a function $f\colon \{0, 1\}^n \longrightarrow \{0, 1\}^n$ of the form $f(g(\mathbf{m})) = \mathbf{x}^\circ\mathbf{m}$. If an input $\mathbf{z} \in \{0, 1\}^n$ is not included in the codomain of the function g, then the output of the function f is assigned by generating a signal, $\mathbf{v} = [v_1, v_2, \ldots, v_n]$, from a random number generator. The generator can produce binary digits, 0, 1, through equal probability sampling. $\Pr\{v_i = 0\} = \Pr\{v_i = 1\} = (1/2)$, where $i \in \mathcal{N}$. Let $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n]$ be a new given signal, and following theorem describes how to recognize the signal $\mathbf{x}$.*

$$\zeta = \frac{1}{2^n} \sum_{\mathbf{m} \in \{0,1\}^n} Z(f(\tilde{\mathbf{x}}^\circ(1 - \mathbf{m})), \tilde{\mathbf{x}}^\circ\mathbf{m}), \tag{5}$$

where

$$Z(\mathbf{a}, \mathbf{b}) = \begin{cases} 1, & \text{if } \mathbf{a} = \mathbf{b}, \\ 0, & \text{if } \mathbf{a} \neq \mathbf{b}. \end{cases} \tag{6}$$

If $\tilde{\mathbf{x}} \neq \mathbf{x}$, then

$$\Pr\{\zeta = 1\} = 0, \tag{7}$$

$$\Pr\{\zeta = 0\} > \left(1 - \frac{1}{2^n}\right)^{2^n} \longrightarrow \frac{1}{e} \approx 0.3679. \tag{8}$$

If $\tilde{\mathbf{x}} = \mathbf{x}$, then

$$\begin{aligned} \zeta &= \frac{1}{2^n} \sum_{\mathbf{m} \in \{0,1\}^n} Z(f(\tilde{\mathbf{x}}^\circ(1 - \mathbf{m})), \tilde{\mathbf{x}}^\circ\mathbf{m}) = \frac{1}{2^n} \sum_{\mathbf{m} \in \{0,1\}^n} Z(f(\mathbf{x}^\circ(1 - \mathbf{m})), \mathbf{x}^\circ\mathbf{m}) \\ &= \frac{1}{2^n} \sum_{\mathbf{m} \in \{0,1\}^n} Z(f(g(\mathbf{m})), \mathbf{x}^\circ\mathbf{m}) = \frac{1}{2^n} \sum_{\mathbf{m} \in \{0,1\}^n} Z(\mathbf{x}^\circ\mathbf{m}, \mathbf{x}^\circ\mathbf{m}) \\ &= 1. \end{aligned} \tag{9}$$

The proof of Theorem 1 is given in Appendix.

According to equations (7)–(9), recognizing the signal $\mathbf{x}$ with the function f is feasible. When a point can help the function f to retrieve the signal $\mathbf{x}$, the point is called an *evoked point*. Let

$$\mathbf{D} \triangleq \{\mathbf{x}^\circ(1 - \mathbf{m})\colon \mathbf{m} \in \{0, 1\}^n\}. \tag{10}$$

There are $|\mathbf{D}|$ *evoked points*, where $|\mathbf{D}| = 2^{|\mathbf{x}|_{l_1}} = 2^{x_1 + x_2 + \cdots + x_n}$. If $\Pr\{x_i = 1\} = (1/2)$, then the expected value of $|\mathbf{D}|$ is $2^{(n/2)}$.

The function f can retrieve the signal $\mathbf{x}$ from the *evoked point*, $\mathbf{h} = \tilde{\mathbf{x}}^\circ(1 - \mathbf{m})$, in the current input signal $\tilde{\mathbf{x}}$ since the signal $\mathbf{x}$ can be expressed as $\mathbf{h} + f(\mathbf{h})$. By comparing $f(\mathbf{h})$ with $\tilde{\mathbf{x}}^\circ\mathbf{m}$, identifying whether $\tilde{\mathbf{x}}$ is the same as $\mathbf{x}$ or not is feasible. If there exists a mask $\mathbf{m} \in \{0, 1\}^n$ that can make $f(\tilde{\mathbf{x}}^\circ(1 - \mathbf{m})) = \tilde{\mathbf{x}}^\circ\mathbf{m}$, then $\Pr\{\tilde{\mathbf{x}} = \mathbf{x}\} = 1 - (1/2^n)$. If $\tilde{\mathbf{x}} = \mathbf{x}$, then each mask $\mathbf{m} \in \{0, 1\}^n$ can make $f(\tilde{\mathbf{x}}^\circ(1 - \mathbf{m})) = \tilde{\mathbf{x}}^\circ\mathbf{m}$.

In a similar way of recognizing the signal $\mathbf{x}$ (Theorem 1), identifying whether the new given signal $\tilde{\mathbf{x}}$ is in a set $\mathbf{X} \triangleq \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\}$ or not is also feasible.

Lemma 1. *Suppose $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}] \in \{0, 1\}^n$. If the signals, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$, satisfy $\mathbf{x}_i \neq \mathbf{x}_j$ for any $i \neq j$, then there exists $p - 1$ positive integer, $\alpha_1, \alpha_2, \ldots, \alpha_{p-1}$, s.t., $[x_{i,\alpha_1}, x_{i,\alpha_2}, \ldots, x_{i,\alpha_{p-1}}] \neq [x_{j,\alpha_1}, x_{j,\alpha_2}, \ldots, x_{j,\alpha_{p-1}}]$ for any $i \neq j$.*

The proof of Lemma 1 is given in the Appendix.

Theorem 2. *Suppose the signals, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$, satisfy $\mathbf{x}_i \neq \mathbf{x}_j$ for any $i \neq j$ where $\mathbf{x}_i$ is a constant. Without loss of generality,*

assume that $[x_{i,1}, x_{i,2}, \ldots, x_{i,p-1}] \neq [x_{j,1}, x_{j,2}, \ldots, x_{j,p-1}]$ *for any* $i \neq j$. *Let*

$$\mathbf{B}_i = \mathbf{D}_i \setminus \bigcup_{j \in \mathscr{P}, j \neq i} \mathbf{D}_i \cap \mathbf{D}_j, \tag{11}$$

where

$$\mathbf{D}_i \triangleq \{\mathbf{x}_i \circ (1 - \mathbf{m}) \colon \mathbf{m} \in \{0,1\}^n\}, \quad i \in \mathscr{P} \triangleq \{1, \ldots, p\}. \tag{12}$$

If $[x_{i,1}, x_{i,2}, \ldots, x_{i,p-1}] \neq [0, 0, \ldots, 0]$, then the cardinality of $\mathbf{B}_i$ satisfies that

$$|\mathbf{B}_i| \geq 2^{x_{i,p} + x_{i,p+1} + \cdots + x_{i,n}}. \tag{13}$$

If $[x_{i,1}, x_{i,2}, \ldots, x_{i,p-1}] = [0, 0, \ldots, 0]$, *then*

$$|\mathbf{B}_i| = 2^{x_{i,p} + x_{i,p+1} + \cdots + x_{i,n}} - 1. \tag{14}$$

The proof of Theorem 2 is given in Appendix.

Assume that any signal in set $\mathbf{X}$ does not equal $[0, 0, \ldots, 0]$, i.e., $|\mathbf{x}_i|_{l_1} = x_{i,1} + x_{i,2} + \cdots + x_{i,n} \geq 1$ for each $i \in \mathscr{P}$. Then, $|\mathbf{B}_i| \geq 1$ where $i \in \mathscr{P}$. If $\Pr\{x_i = 1\} = (1/2)$, then the expected value of $|\mathbf{B}_i|$ is $2^{((n-p+1)/2)}$.

When a new signal $\tilde{\mathbf{x}}$ appears, it is possible to identify whether $\tilde{\mathbf{x}}$ is in the set $\mathbf{X}$ or not with a function $f \colon \{0,1\}^n \longrightarrow \{0,1\}^n$. Construct the function f of the form

$$f(\mathbf{x}_i \circ (1 - \mathbf{m})) = \mathbf{x}_i \circ \mathbf{m}, \tag{15}$$

for any $\mathbf{x}_i \circ (1 - \mathbf{m}) \in \mathbf{B}_i$, $i \in \mathscr{P}$. If an input $\mathbf{z} \in \{0,1\}^n$ is not included $\mathbf{B}_1 \cup \cdots \cup \mathbf{B}_p$, then the output of the function f is assigned by generating a signal from a random number generator. If there exists a mask $\mathbf{m} \in \{0,1\}^n$ that can make $f(\tilde{\mathbf{x}} \circ (1 - \mathbf{m})) = \tilde{\mathbf{x}} \circ \mathbf{m}$, then $\Pr\{\tilde{\mathbf{x}} \in \mathbf{X}\} = 1 - (1/2^n)$. If $\tilde{\mathbf{x}} \in \mathbf{X}$ (for example $\tilde{\mathbf{x}} = \mathbf{x}_j$), then there exists at least one mask $\mathbf{m} \in \{0,1\}^n$ that can make $f(\tilde{\mathbf{x}} \circ (1 - \mathbf{m})) = \tilde{\mathbf{x}} \circ \mathbf{m}$ since $|\mathbf{B}_j| \geq 1$.

To identify whether a new given signal $\tilde{\mathbf{x}}$ has the same label as $\mathbf{X}$, the distance/similarity between $\tilde{\mathbf{x}}$ and $\mathbf{X}$ can be used, such as

$$d(\tilde{\mathbf{x}}, \mathbf{X}) = \inf\{d(\tilde{\mathbf{x}}, \mathbf{x}) | \mathbf{x} \in \mathbf{X}\}. \tag{16}$$

Without memorizing any element in $\mathbf{X}$, it is also possible to estimate the similarity by using the function f.

Let

$$\begin{aligned} \tilde{\mathbf{D}} &= \{\tilde{\mathbf{x}} \circ (1 - \mathbf{m}) \colon \mathbf{m} \in \{0,1\}^n\}, \\ \mathbf{B} &= \mathbf{B}_1 \cup \cdots \cup \mathbf{B}_p. \end{aligned} \tag{17}$$

If $\tilde{\mathbf{D}} \cap \mathbf{B}_i \neq \varnothing$, then there exist $|\tilde{\mathbf{D}} \cap \mathbf{B}_i|$ evoked points. Let $q_i = |\tilde{\mathbf{D}} \cap \mathbf{B}_i|$. Each evoked point, $\mathbf{h}_{i,j}$, is corresponding to a mask, $\mathbf{m}_{i,j}$.

$$\mathbf{h}_{i,j} = \tilde{\mathbf{x}} \circ (1 - \mathbf{m}_{\mathrm{i,j}}) = \mathbf{x}_{\mathrm{i}} \circ (1 - \mathbf{m}_{\mathrm{i,j}}), \tag{18}$$

where $j \in \{1, 2, \ldots, p_i\}$. The evoked points and the masks can be used to retrieve $\mathbf{x}_i$. With the function f, $\mathbf{x}_i$ can be gotten by

$$\begin{aligned} &\tilde{\mathbf{x}} \circ (1 - \mathbf{m}_{\mathrm{i,j}}) + \mathrm{f}(\tilde{\mathbf{x}} \circ (1 - \mathbf{m}_{\mathrm{i,j}})) \\ &= \mathbf{x}_i \circ (1 - \mathbf{m}_{\mathrm{i,j}}) + \mathrm{f}(\mathbf{x}_{\mathrm{i}} \circ (1 - \mathbf{m}_{\mathrm{i,j}})) \\ &= \mathbf{x}_i \circ (1 - \mathbf{m}_{\mathrm{i,j}}) + \mathbf{x}_{\mathrm{i}} \circ \mathbf{m}_{\mathrm{i,j}} \\ &= \mathbf{x}_i. \end{aligned} \tag{19}$$

The similarity between $\tilde{\mathbf{x}}$ and $\mathbf{X}$ can be estimated by

$$\inf\{d(\tilde{\mathbf{x}}, \tilde{\mathbf{x}} \circ (1 - \mathbf{m}) + f(\tilde{\mathbf{x}} \circ (1 - \mathbf{m}))) | \mathbf{m} \in \{0,1\}^n\}. \tag{20}$$

The process of similarity estimation is influenced by $\tilde{\mathbf{x}} \circ (1 - \mathbf{m}) \notin \mathbf{B}$. If $\tilde{\mathbf{x}}$ has the same meaning as element in $\mathbf{X}$, then $\tilde{\mathbf{x}}$ has to overcome the influence. The drawback of this estimation is that the algorithm might not traverse all elements in $\mathbf{X}$. If $\tilde{\mathbf{D}} \cap \mathbf{B}_i = \varnothing$, then there are no evoked points that can help us to retrieve $\mathbf{x}_i$. However, the advantage is that intentionally recollecting all elements in $\mathbf{X}$ is not necessary.

By training l functions $f_1, \ldots, f_l$ to predict the masked part of elements in their respective instance sets, i.e., $\mathbf{X}_1, \ldots, \mathbf{X}_l$, recognizing a new given signal $\tilde{\mathbf{x}}$ as one category is feasible. Let $\mathbf{X}_i \triangleq \{\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,p_i}\}$, $\mathbf{D}_{i,j} \triangleq \{\mathbf{x}_{i,j} \circ (1 - \mathbf{m}) \colon \mathbf{m} \in \{0,1\}^n\}$, and

$$\mathbf{B}_{i,j} = \mathbf{D}_{i,j} \setminus \left(\bigcup_{k \in \{1, \ldots, p_i\}, k \neq j} \mathbf{D}_{i,j} \cap \mathbf{D}_{i,k} \right), \tag{21}$$

where $j \in \{1, \ldots, p_i\}$, $i \in \{1, \ldots, l\}$. Function f_i satisfies

$$f_i(\mathbf{x}_{i,j} \circ (1 - \mathbf{m})) = \mathbf{x}_{i,j} \circ \mathbf{m}, \tag{22}$$

for any $\mathbf{x}_{i,j} \circ (1 - \mathbf{m}) \in \mathbf{B}_{\mathrm{i,j}}$, $j \in \{1, \ldots, p_i\}$. If an input $\mathbf{z} \in \{0,1\}^n$ is not included $\mathbf{B}_{i,1} \cup \cdots \cup \mathbf{B}_{i,p_i}$, then the output of the function f_i is assigned by generating a signal from a random number generator.

When a signal $\tilde{\mathbf{x}}$ is to be labeled, the similarity rule is a natural choice. To reduce the influence on the analysis of similarity, replace the infimum with average. The similarity between $\tilde{\mathbf{x}}$ and $\mathbf{X}_i$ can be estimated by

$$s_i(\tilde{\mathbf{x}}) \triangleq \frac{1}{2^n} \sum_{\mathbf{m} \in \{0,1\}^n} d(\tilde{\mathbf{x}}, \tilde{\mathbf{x}} \circ (1 - \mathbf{m}) + \mathrm{f_i}(\tilde{\mathbf{x}} \circ (1 - \mathbf{m}))). \tag{23}$$

The smaller the similarity is, the more likely they are in the same category and have the same label. The above recognition model is presented in Algorithm 1, which is called the Implicit Recognition Model.

In summary, the Implicit Recognition Model uses similarity comparison to do the recognition. But intentionally recalling any labeled signals is not necessary. The focus is not on the past, but on the current input. The evoked points are objective features, which can help us to retrieve labeled signals from the current input. Both *evoked points* and retrieved signals have explicit physical meaning in the real world, which are objective pieces of evidence to support the judgment.

4. Experiment

In the application, the Mixed National Institute of Standards and Technology (MNIST) database is used to verify the

input: $\mathbf{X}_i = \left\{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,p_i}\right\}$ is a training instance set and $\mathbf{y}_i$ is the known label of $\mathbf{X}_i$; f_i, labeled $\mathbf{y}_i$, is an approximator and can be trained; $i \in \mathscr{L}$; $\tilde{\mathbf{x}}$ is a testing instance.

* Cognitive Process *

(1) **for** each $i \in \mathscr{L}$ **do**
(2) **repeat**
(3) Observing a signal in $\mathbf{X}_i$;
(4) Training the prediction function f_i to predict the masked parts of the signal;
(5) **until** For each signal $\mathbf{x}_{i,k} \in \mathbf{X}_i$ and any $j \neq i$, $s_i(\mathbf{x}_{i,k}) < s_j(\mathbf{x}_{i,k})$.
(6) **end for**
(7) **return** Prediction functions $f_1, \ldots, f_l$ labeled with $\mathbf{y}_1, \ldots, \mathbf{y}_l$ respectively.

* Recognition Process *

(1) **for** each $i \in \mathscr{L}$ **do**
(2) Estimating the similarity between testing signal $\tilde{\mathbf{x}}$ and instance set $\mathbf{X}_i$;
(3) Let $\varsigma_i = s_i(\tilde{\mathbf{x}})$;
(4) **end for**
(5) Assigning the image $\tilde{\mathbf{x}}$ to the class of its highest similarity;
(6) $\hat{l} = \operatorname{argmin}_i \varsigma_i$;
(7) **return** $\mathbf{y}_{\hat{l}}$.

Algorithm 1: Implicit Recognition Model.

Implicit Recognition Model. The MNIST database is one of the famous image classification benchmarks [26]. There are 60,000 instances in the training set and 10,000 instances in the testing set. The database is created by remixing the samples of NIST's original data sets. The black and white images from NIST are normalized to fit into a 28×28 pixel bounding box and antialiased, which introduces gray-scale levels [27]. The first 500 elements of the training set are used for training. All of the testing instances are used for testing.

$\mathbf{X}_i$ is labeled i and only contains training images with label i where $i = 0, 1, \ldots, 9$. Among the first 500 training images, there are $p_0 = 62$ training images labeled 0, $p_1 = 51$ training images labeled 1, and so on. $|\mathbf{X}_0| + \cdots + |\mathbf{X}_9| = 500$. All the 500 images are different. $\mathbf{X}_i \cap \mathbf{X}_j = \varnothing$ for any $i \neq j$.

In the process of recognition, an image in the testing set is recognized as the class of the highest similarity. To execute Algorithm 1, a distance function d is redefined first. Cosine distance is a usual measure of similarity in machine learning [28, 29]. Given two vectors $\mathbf{a}$ and $\mathbf{b}$, the cosine similarity $\cos(\theta)$ is represented by

$$\cos(\theta) = \frac{\mathbf{ab}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}, \tag{24}$$

where a_i and b_i are components of the vectors $\mathbf{a}$ and $\mathbf{b}$, respectively. The angle $\theta \in [0, \pi]$ is used as the distance between two images in this application. Therefore,

$$d(\mathbf{a}, \mathbf{b}) = \arg\cos\left(\frac{\mathbf{ab}}{\|\mathbf{a}\|\|\mathbf{b}\|}\right). \tag{25}$$

With the artificial neural network, the implicit memory is simulated to execute the Implicit Recognition Model (Algorithm 1). The main process of Algorithm 1 is to train 10 approximators of the functions $f_0, f_1, \ldots, f_9$. Then, f_i can be used to retrieve images in $\mathbf{X}_i$ from the evoked points of the current input image $\tilde{\mathbf{x}}$ and estimate the similarity between $\tilde{\mathbf{x}}$ and $\mathbf{X}_i$.

To imitate the human recognition process, 10 ANNs are trained in two steps. The first step is to train a base neural network to complete all training images by filling in missing regions of a rectangular shape. This step does not use labels. In order to finally generate the specific function f_i, the second step is to add *routing layers* to the trained base neural network. With supervision, the *routing layers* are trained on the instance set $\mathbf{X}_i$. Routing neural networks are used to approximate the specific functions.

4.1. Architecture of ANN. A fully convolutional neural network is modified to be the base neural network. Refer to [30] for more details of the original fully convolutional neural network. The modified network parameters are given in Tables 3–6. Behind each convolution layer, except the last one, there is a Rectified Linear Unit (ReLU) layer. The output layer consists of a convolutional layer with a sigmoid function instead of a ReLU layer to normalize the output to the [0, 1] range. "Outputs" refers to the number of output channels for the output of the layer (Tables 3–6). Fully connected (FC) layers refer to the standard neural network layers. The output layer of discriminator consists of a fully connected layer with a sigmoid transfer layer. The discriminator outputs the probability that an input image came from real images rather than the completion network.

Routing layers are added to the trained base neural network to generate an approximator of the specific function f_i. Figure 8 shows the architecture of a routing neural network. Three routing layers are inserted in front of each network layer. The routing layer performs element-wise multiplication between the input $\mathbf{x} = [x_1, \ldots, x_n]$ and the routing weights $\mathbf{w} = [w_1, \ldots, w_n]$, which can be represented as $\mathbf{x}^{\circ}\mathbf{w} = [\mathrm{x}_1\mathrm{w}_1, \ldots, \mathrm{x}_\mathrm{n}\mathrm{w}_\mathrm{n}]$. The initial value of routing weight $\mathbf{w}$ is set to $1 = [1, \ldots, 1]$. The role of routing layers is to keep $f_i(\mathbf{x}_{i,k}{}^{\circ}(1 - \mathbf{m}))^{\circ}\mathbf{m} + \mathbf{x}_{\mathrm{i,k}}{}^{\circ}(1 - \mathbf{m})$ similar to $\mathbf{x}_{i,k}$, but the distance between $f_j(\mathbf{x}_{i,k}{}^{\circ}(1 - \mathbf{m}))^{\circ}\mathbf{m} + \mathbf{x}_{\mathrm{i,k}}{}^{\circ}(1 - \mathbf{m})$ and $\mathbf{x}_{i,k}$ is becoming far with the training, for any $j \neq i$, $\mathbf{x}_{i,k} \in \mathbf{X}_i$.

Table 3: Architecture of base neural network.

Type	Kernel	Dilation (μ)	Stride	Outputs
Conv.	3×3	1	1×1	14
Conv.	2×2	1	1×1	28
Conv.	2×2	1	1×1	28
Conv.	2×2	1	1×1	56
Conv.	2×2	1	1×1	56
Conv.	2×2	1	1×1	56
Dilated conv.	2×2	2	1×1	56
Dilated conv.	2×2	2	1×1	56
Dilated conv.	2×2	2	1×1	56
Dilated conv.	2×2	2	1×1	56
Conv.	2×2	1	1×1	56
Conv.	2×2	1	1×1	56
Deconv.	2×2	1	1×1	28
Conv.	2×2	1	1×1	28
Deconv.	2×2	1	1×1	14
Conv.	2×2	1	1×1	7
Output	2×2	1	1×1	1

Table 4: Architecture of local discriminator.

Type	Kernel	Stride	Outputs
Conv.	2×2	1×1	14
Conv.	2×2	1×1	28
Conv.	2×2	1×1	56
Conv.	2×2	1×1	112
FC	—	—	112

Table 5: Architecture of global discriminator.

Type	Kernel	Stride	Outputs
Conv.	2×2	1×1	14
Conv.	2×2	1×1	28
Conv.	2×2	1×1	56
Conv.	2×2	1×1	112
Conv.	2×2	1×1	112
FC	—	—	112

Table 6: Concatenation layer of discriminator.

Type	Kernel	Stride	Outputs
Concat	—	—	224
FC	—	—	1

4.2. Training Method of ANN. While the base neural network is training, global and local context discriminators are trained to distinguish real images from completed ones. The base neural network is trained to complete images by filling in masked regions of a rectangle. The training operation on the mask set $\mathbf{M} = \{\mathbf{m}: \mathbf{m} \in \{0,1\}^{28\times 28}\}$ is too large to calculate. A subset (denoted $\mathbf{M}'$) of $\mathbf{M}$ is selected with a rectangle masked part of 7 to 14 for the width and height.

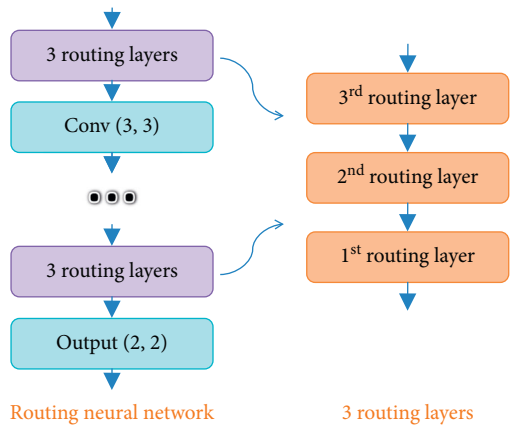


Figure 8: Network architectures of the routing neural network and 3 routing layers.

$$M' = \left\{ \mathbf{m} | \mathbf{m} = \underbrace{0^{(28)}; \ldots ; 0^{(28)};}_{a_0} \begin{matrix} 0^{(b_0)}, 1^{(b_1)}, 0^{(b_2)}; \\ \vdots \\ 0^{(b_0)}, 1^{(b_1)}, 0^{(b_2)}; \end{matrix} \right\} a_1 \underbrace{0^{(28)}; \ldots ; 0^{(28)}}_{a_2}, a_0 + a_1 + a_2 = 28, b_0 + b_1 + b_2 = 28, a_1, b_1 \in \{7, \ldots, 14\}, \quad (26)$$

where

$$0^{(k)} = \underbrace{[0, \ldots, 0]}_{k}, \quad 1^{(k)} = \underbrace{[1, \ldots, 1]}_{k}. \quad (27)$$

All the 500 training images (denoted $\mathbf{X}$) are used to train the base neural network.

When the training of the base neural network is finished, routing layers are added to generate an approximator of specific prediction function f_i. While the routing layers are training, the parameters of the base neural network remain unchanged. Ten digit labels/categories correspond to 10 approximators of prediction functions. Each prediction function f_i has a unique set of routing parameters.

The training method for routing layers is the same as the base neural network. The routing neural network is also trained to complete images by filling in masked regions of a rectangle. However, only the instance set $\mathbf{X}_i$ is used to train routing parameters of approximator of f_i. The three routing layers are trained one after another. While one routing layer is training, the other two routing layers remain unchanged. While $s_i(\mathbf{x}_{i,k})$ is becoming less than $s_j(\mathbf{x}_{i,k})$ for each signal $\mathbf{x}_{i,k} \in \mathbf{X}_i$ and any $j \neq i$, the cognitive process finishes. The trained routing neural network is an approximator of the specific prediction tool/function f_i.

4.3. Experimental Results. The neural network models are created by TensorFlow 1.8.0. The learning rate of the Adam optimizer is 10^{-3}. A batch size of one image is used for training. First, the base neural network is trained for 100 iterations. Then both the discriminator and base neural network are jointly trained. In each iteration, all 500 training images are shuffled and traversed. For each image, a masked rectangle region is randomly selected to train the neural networks. By 1000 iterations, the base neural network can predict the masked part (Figure 9).

Then, keep the base neural network unchanged, and add routing layers. The first routing layers are trained for 260 iterations and then remain unchanged. The second routing layers are also trained for 260 iterations and then remain unchanged. The approximation of a specific prediction function finishes when the third routing layers are also trained for 260 iterations.

A set of routing parameters correspond to a specific prediction function. Each approximator of functions $f_0, \ldots, f_9$ is corresponding to a routing neural network with a particular set of routing parameters. The approximator of function f_i can only work on the training set $\mathbf{X}_i$. Figure 10 gives an example. The mean distance $\overline{d_{i,f_i}}$ is less than $\overline{d_{i,f_j}}$ for each $j \neq i$ where $\overline{d_{i,f_j}} \triangleq (1/p_i) \sum_{k=1}^{p_i} s_j(\mathbf{x}_{i,k})$ (Figure 11).

With the approximator of the prediction function f_i, the algorithm can estimate the similarity between a testing image $\tilde{\mathbf{x}}$ and the instance set $\mathbf{X}_i$. A subset of M' (denoted $\mathbf{M}_r$) is used to estimate the similarity. Equations (23) and (25) are modified by

$$s_i(\tilde{\mathbf{x}}) \triangleq \frac{1}{|\mathbf{M}_r|} \sum_{\mathbf{m} \in \mathbf{M}_r} d(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^\circ(1 - \mathbf{m}) + \mathrm{f_i}(\tilde{\mathbf{x}}^\circ(1 - \mathbf{m}))^\circ \mathbf{m}). \quad (28)$$

30 masks are randomly generated, and $|\mathbf{M}_r| = 30$. The masks remain unchanged when the algorithm calculates $\varsigma_0, \ldots, \varsigma_9$. Finally, the image $\tilde{\mathbf{x}}$ can be recognized as the digit $\hat{l}$ satisfying that $\varsigma_{\hat{l}} \leq \varsigma_i$ for any $i \in \{0, 1, \ldots, 9\}$.

The masked part is the key for doing the similarity comparison. If the mask's size is too small, $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}^\circ(1 - \mathbf{m}) + \mathrm{f}(\tilde{\mathbf{x}}^\circ(1 - \mathbf{m}))^\circ \mathbf{m}$ are mostly the same. There is no enough data to demonstrate the similarity. The main cause of high similarity should be from the accuracy of prediction (i.e., $f(\tilde{\mathbf{x}}^\circ(1 - \mathbf{m}))^\circ \mathbf{m}$ is similar to $\tilde{\mathbf{x}}^\circ \mathbf{m}$), not the small mask

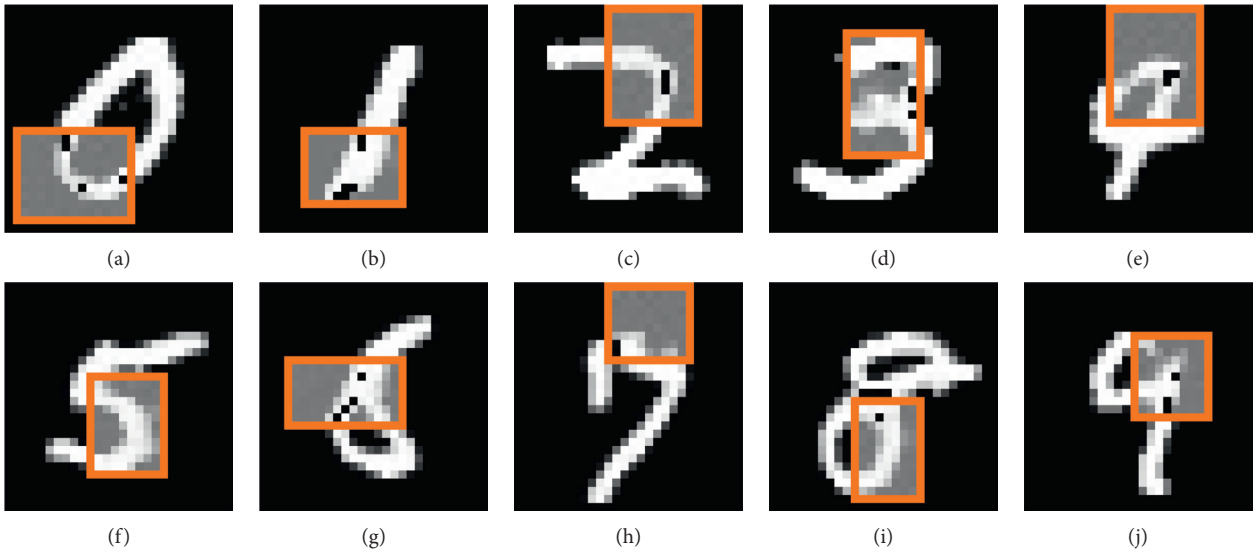


Figure 9: Prediction result by a base neural network on a training image with a label of (a) 0, (b) 1, (c) 2, (d) 3, (e) 4, (f) 5, (g) 6, (h) 7, (i) 8, and (j) 9 using randomly generated masks. The mask covers up a rectangular part of the image, and the predicted result is in the box.

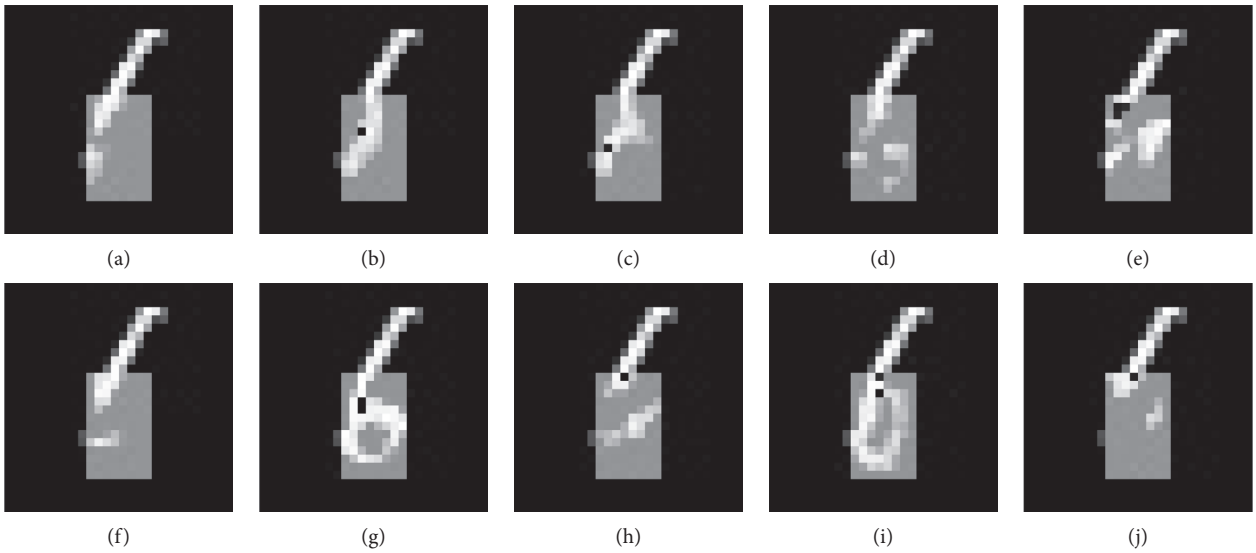


Figure 10: Prediction result by a routing neural network of (a)f_0, (b)f_1, (c)f_2, (d)f_3, (e)f_4, (f)f_5, (g)f_6, (h)f_7, (i)f_8, and (j)f_9 on a training image in $\mathbf{X}_6$ using a randomly generated mask.

window (i.e., $|\mathbf{m}|_{l_1}$ is small). An appropriate size can ensure the estimation results would be more comparable and credible. For a similarity comparison, the mask's window size cannot be too small. Meanwhile, the size cannot be too large to train ANN. The bigger the size is, the harder the masked parts are to be predicted.

Using the Implicit Recognition Model (Algorithm 1), the correct recognition probability approaches 100% on the training set (contains 500 training images). "Digit 4" and "digit 9" are easily to be confused. 80.44% recognition results of the Implicit Recognition Model are the same as the labels, and the confusion matrix is shown in Figure 12. Experimental results show the efficiency of the proposed Implicit Recognition Model.

The proposed model recognizes image $\tilde{\mathbf{x}}$ as digit $\hat{l}$, because $\tilde{\mathbf{x}}$ is most similar to the instances in $\mathbf{X}_{\hat{l}}$. As the similarity becomes lower, both the recognition precision of the Implicit Recognition Model and the one nearest neighbor algorithm (Explicit Recognition Model) decrease while rotating the testing images (Figure 13). Even using the testing image $\tilde{\mathbf{x}}$ itself to do the rotation similarity comparison, the distance $d(\tilde{\mathbf{x}}, r(\tilde{\mathbf{x}}, \alpha))$ becomes further still (Figure 14), where $r(\tilde{\mathbf{x}}, \alpha)$ represents that image $\tilde{\mathbf{x}}$ rotates α degrees. While the rotation angle is around 90 and 270 degrees, the distance approaches 1.22 radians, which is the mean distance between a noise image and the testing image. The pixel of testing image takes integer values between 0 and 255. The pixel of noise image obeys discrete uniform distribution on

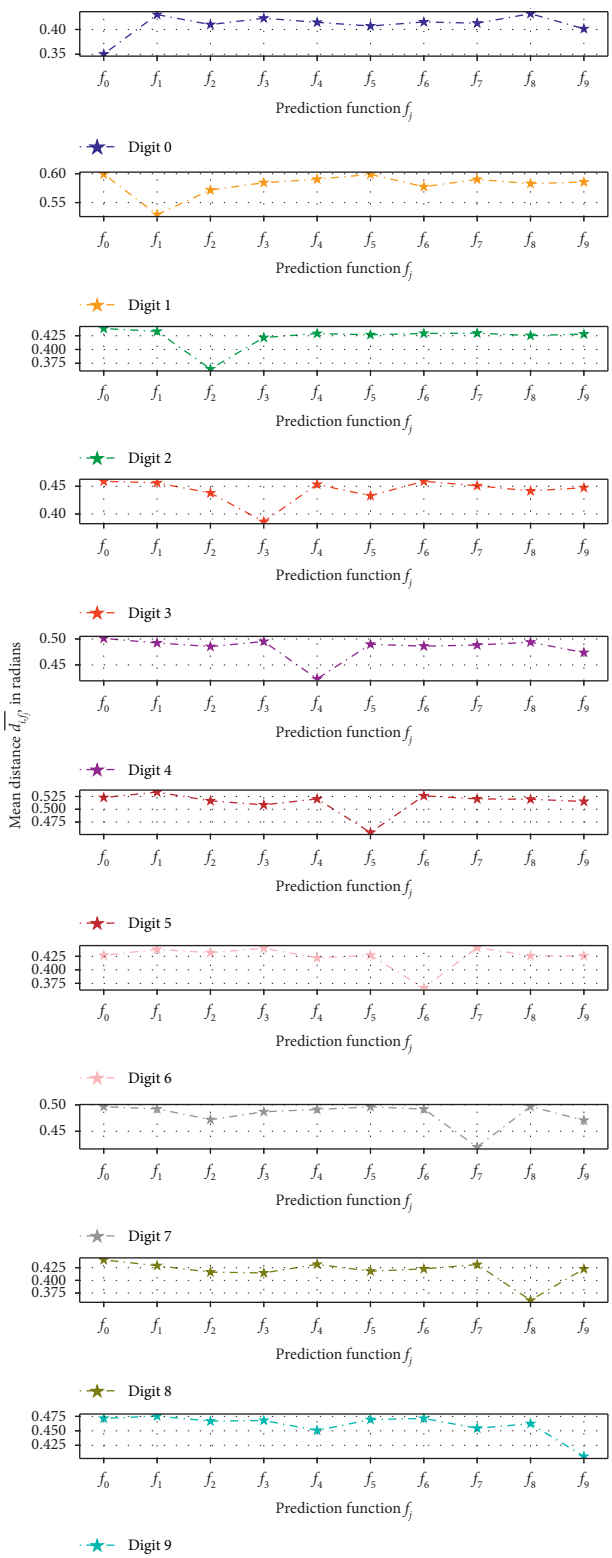


FIGURE 11: Mean distance $\overline{d_{i,f_j}}$ for each f_j on the training set.

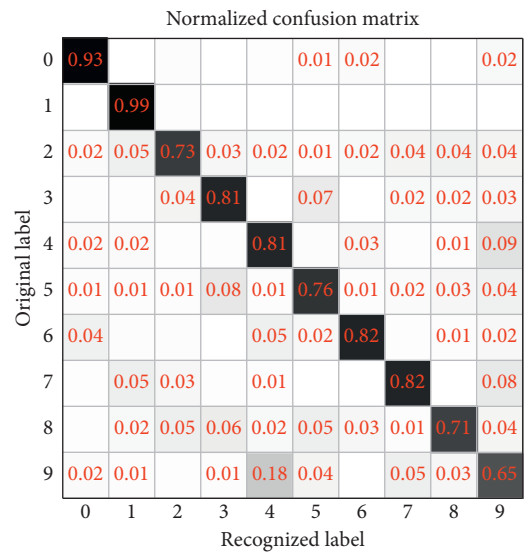


FIGURE 12: Performance of Implicit Recognition Model. Each column of the matrix represents the instances in an original class while each row represents the instances in a recognized class. The figure is accurate to two decimal places.

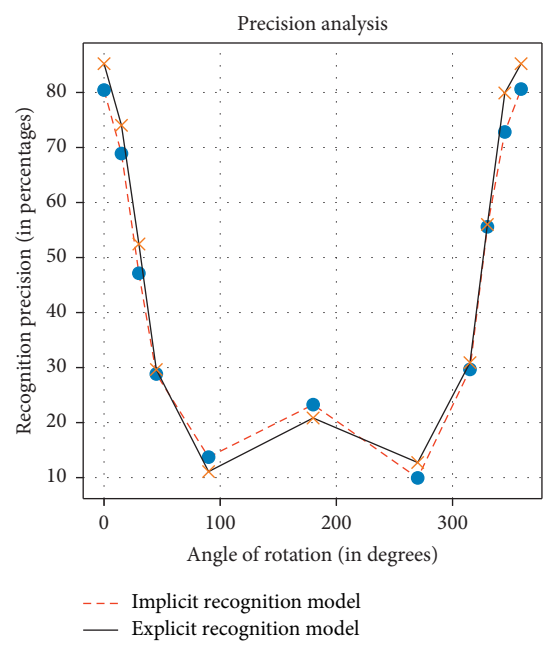


FIGURE 13: Recognition precision when the testing images rotate 0, 15, 30, 45, 90, 180, 270, 315, 330, 345, and 359 degrees, respectively.

the integers $0, 1, 2, \ldots, 255$. In Figures 13 and 14, the curves show W-shape and M-shape, because the images labeled 1 are similar to their rotated images when the rotation angle is around 180 degrees. Rotation of image can cause similarity changes. Therefore, a rotated image can also be used to define a new category if the similarity change is big enough. For example, the main difference between "W" and "M" is the direction of the opening (Figures 13 and 14).

Furthermore, the proposed algorithm is robust while adding random noise $\mathbf{n}$ to the testing images. Suppose that $\mathbf{n} = [n_1, \ldots, n_{784}]$ where n_i is an independent random variable. Noise point n_i obeys discrete uniform distribution on the integers $\beta, \beta+1, \beta+2, \ldots, \gamma$, where β and γ restrict the value range $(0 \le \beta \le \gamma \le 255)$. $\{\beta, \gamma\}$ represents the noise level. The bigger the values, the higher the noise level. The

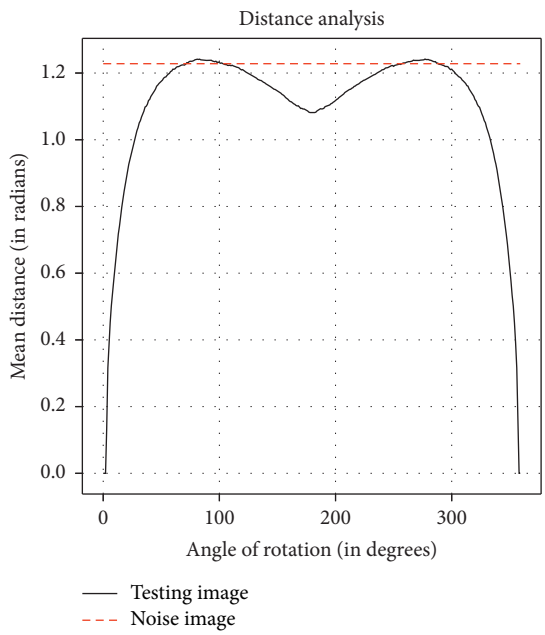


Figure 14: Rotation similarity comparison. Solid line represents the mean distance between a rotated image and its original image in the testing set. Dashed line represents the mean distance between a noise image and the testing image.

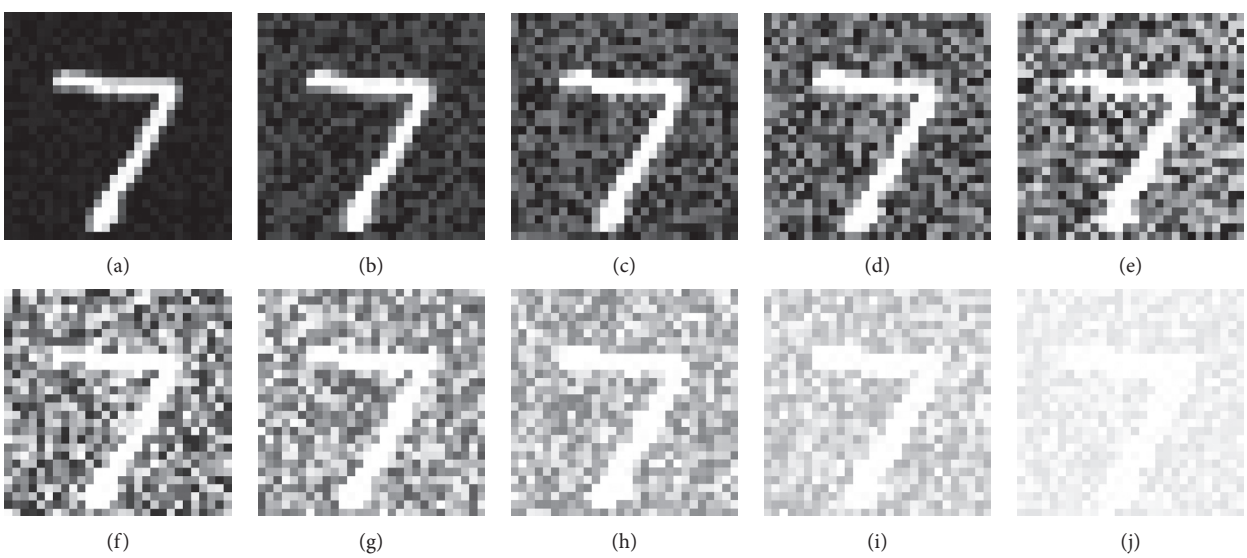


Figure 15: Testing image is disturbed at a noise level of (a) {0, 25}, (b) {0, 76}, (c) {0, 127}, (d) {0, 178}, (e) {0, 229}, (f) {25, 255}, (g) {76, 255}, (h) {127, 255}, (i) {178, 255}, and (j) {229, 255}.

testing images $\tilde{\mathbf{x}} = [\tilde{x}_1, \ldots, \tilde{x}_{784}]$ are disturbed by noise $\mathbf{n}$ (Figure 15) to test the robustness of algorithms. The image in noise is denoted as $\tilde{\mathbf{x}}_{\text{noise}} = [T(\tilde{x}_1 + n_1), \ldots, T(\tilde{x}_{784} + n_{784})]$ where

$$T(x_0 + x_1) = \begin{cases} x_0 + x_1, & \text{if } x_0 + x_1 \leq 255, \\ 255, & \text{if } x_0 + x_1 > 255. \end{cases} \tag{29}$$

Using PyTorch 1.5.1, the traditional neural network models are modified to do the recognition, too. Two linear layers after the output layer are added. In front of each linear layer, there is a Rectified Linear Unit (ReLU) layer. The testing images are resized and repeated along the channel. Then, the stochastic gradient descent optimizer and a batch size of 32 images are used for training.

By training 1000 iterations, the traditional neural network models can work when the noise level is lower than {0, 127}. However, the performance is not stable with increase of the noise level. The recognition precision of many traditional neural networks drops sharply while the change in the noise level is not big, as shown in Figure 16. However, the performance curve of Alexnet and VGG is smooth, as shown in Figure 17. *Overfitting* cannot provide the reasons behind the phenomenon since the neural networks are black-boxes. The

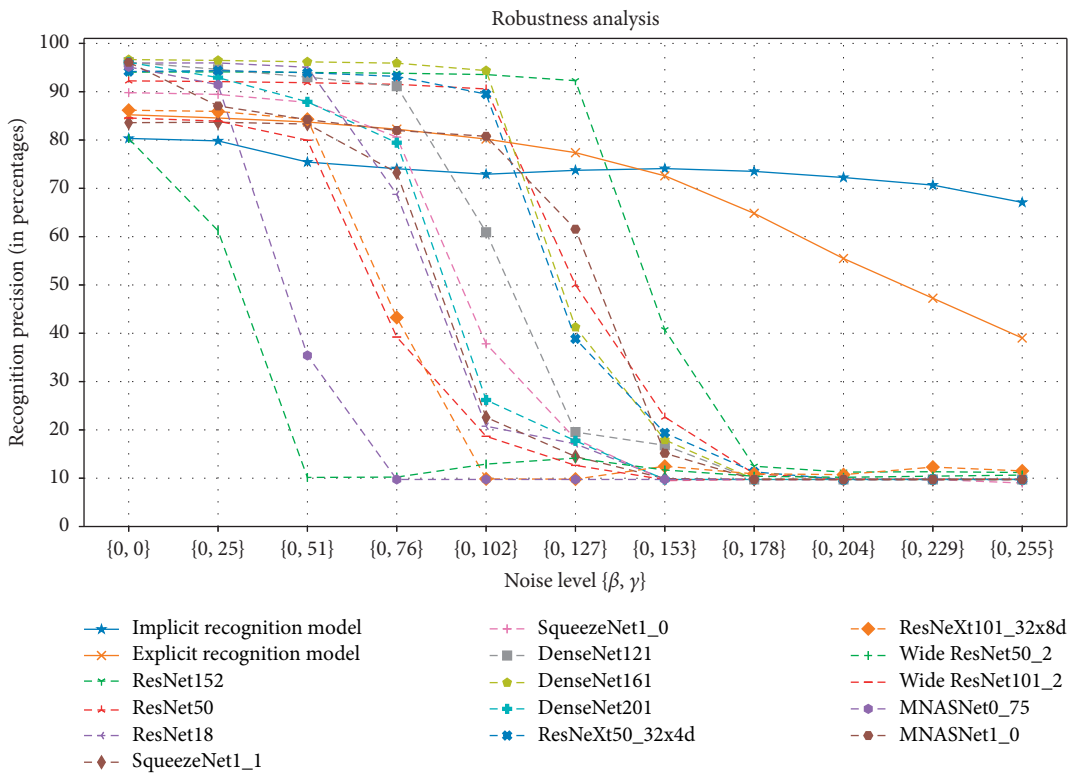


Figure 16: Recognition precision of Implicit Recognition Model, Explicit Recognition Model, ResNet, SqueezeNet, DenseNet, ResNeXt, Wide ResNet, and MNASNet at noise levels {0, 0}, {0, 25}, {0, 51}, {0, 76}, {0, 102}, {0, 127}, {0, 153}, {0, 178}, {0, 204}, {0, 229}, and {0, 255}.

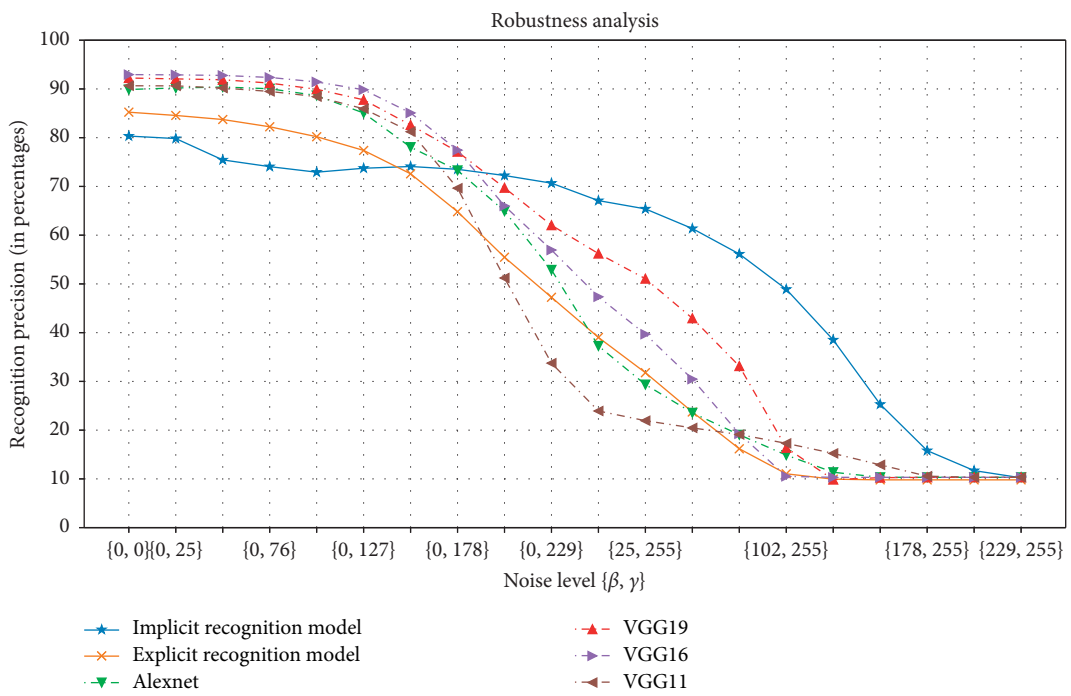


Figure 17: Recognition precision of Implicit Recognition Model, Explicit Recognition Model, Alexnet, and VGG at noise levels {0, 0}, {0, 25}, {0, 51}, {0, 76}, {0, 102}, {0, 127}, {0, 153}, {0, 178}, {0, 204}, {0, 229}, {0, 255}, {25, 255}, {51, 255}, {76, 255}, **{102, 255}**, {127, 255}, {153, 255}, {178, 255}, {204, 255}, and {229, 255}.

human can recognize the digits even when the noise level is $\{229, 255\}$. Therefore, the more straight the performance curve, the closer it is to the human capability. When the noise level is $\{102, 255\}$, the recognition accuracy of the traditional neural networks is less than 20%, but the accuracy of the Implicit Recognition Model is about 50%. The Implicit Recognition Model improves the robustness significantly.

5. Conclusion

Scientists early know the existence of implicit memory. This paper establishes a mathematical model of implicit memory and explains ANN can simulate the model from the view of digital logic circuit design. A trained ANN can be expressed as a function. When a given input is not in the training set, the output of the ANN is hard to control. With the function, this paper proposes a new pattern recognition method, the Implicit Recognition Model. The Implicit Recognition Model works under the similarity rule and has interpretability. Compared to the one nearest neighbor algorithm (Explicit Recognition Model), the Implicit Recognition Model makes similarity comparisons without recalling any instances. The experiment results show the efficiency of the Implicit Recognition Model.

Appendix

A. Proof of Theorem 1

Proof. Construct a function $\tilde{g}$ of the form $\tilde{g}(\mathbf{m}) = \tilde{\mathbf{x}} \circ (1 - \mathbf{m})$. Let

$$\begin{aligned} \mathbf{S} &\triangleq \{g(\mathbf{m}) \colon \mathbf{m} \in \{0,1\}^n\}, \\ \mathbf{T} &\triangleq \{\tilde{g}(\mathbf{m}) \colon \mathbf{m} \in \{0,1\}^n\}. \end{aligned} \tag{A.1}$$

When $\mathbf{m} = [1, 1, \ldots, 1]$, $f(g(\mathbf{m})) = f(\tilde{g}(\mathbf{m})) = f([0, 0, \ldots, 0]) = \mathbf{x} \circ [1, 1, \ldots, 1] = \mathbf{x}$. If $\tilde{\mathbf{x}} \neq \mathbf{x}$, then $Z(f([0, 0, \ldots, 0]), \tilde{\mathbf{x}}) = Z(\mathbf{x}, \tilde{\mathbf{x}}) = 0$. Therefore, $\Pr\{\zeta = 1\} = 0$.

Let $\mathbf{U} = \mathbf{T} \cap \mathbf{S}$ and $\mathbf{V} = \mathbf{T} \setminus \mathbf{S}$. For an arbitrary mask $\mathbf{m}$, $\tilde{g}(\mathbf{m})$ belongs to either $\mathbf{U}$ or $\mathbf{V}$. If $\tilde{g}(\mathbf{m}) \in \mathbf{U}$, then $Z(f(\tilde{g}(\mathbf{m})), \tilde{\mathbf{x}} \circ \mathbf{m}) = 0$. If $\tilde{g}(\mathbf{m}) \in \mathbf{V}$, then

$$\Pr\{Z(f(\tilde{g}(\mathbf{m})), \tilde{\mathbf{x}} \circ \mathbf{m}) = 0\} = 1 - \frac{1}{2^n}. \tag{A.2}$$

If $\tilde{\mathbf{x}} \neq \mathbf{x}$, then $1 \leq |\mathbf{U}| < 2^n$. Because $|\mathbf{U}| + |\mathbf{V}| \leq 2^n$,

$$\begin{aligned} \Pr\{\zeta = 0\} &= \left(1 - \frac{1}{2^n}\right)^{|\mathbf{V}|} \\ &> \left(1 - \frac{1}{2^n}\right)^{|\mathbf{U}| + |\mathbf{V}|} \\ &\geq \left(1 - \frac{1}{2^n}\right)^{2^n} \\ &\longrightarrow \frac{1}{e} \approx 0.3679. \end{aligned} \tag{A.3}$$

□

B. Proof of Lemma 1

Proof. The recurrence method is used to prove this lemma. When $p = 2$, there exists a positive integer, α_1, s.t., $x_{1,\alpha_1} \neq x_{2,\alpha_1}$ since $\mathbf{x}_1 \neq \mathbf{x}_2$.

Suppose the lemma is true when $p = m$. Let $[x_{i,\alpha_1}, \ldots, x_{i,\alpha_{m-1}}] \neq [x_{j,\alpha_1}, \ldots, x_{j,\alpha_{m-1}}]$ for any $i \neq j \in \{1, 2, \ldots, m\}$.

If $[x_{m+1,\alpha_1}, \ldots, x_{m+1,\alpha_{m-1}}] \neq [x_{j,\alpha_1}, \ldots, x_{j,\alpha_{m-1}}]$ for any $j \in \{1, 2, \ldots, m\}$, then α_m can be any positive integer $k \leq n$ where $k \notin \{\alpha_1, \alpha_2, \ldots, \alpha_{m-1}\}$. If $[x_{m+1,\alpha_1}, \ldots, x_{m+1,\alpha_{m-1}}] = [x_{j,\alpha_1}, \ldots, x_{j,\alpha_{m-1}}]$, then there exists a positive integer, α_m, s.t., $x_{m+1,\alpha_m} \neq x_{j,\alpha_m}$ since $\mathbf{x}_{m+1} \neq \mathbf{x}_j$. □

C. Proof of Theorem 2

Proof. Let $\mathbf{S}_i \triangleq \{\mathbf{x}_i \circ (1 - [0^{(p-1)}, \mathbf{m}]) \colon \mathbf{m} \in \{0,1\}^{n-p+1}\}$, where

$$0^{(p-1)} \triangleq \underbrace{[0, \ldots, 0]}_{p-1}. \tag{C.1}$$

$\mathbf{S}_i \in \mathbf{B}_i$ since $[x_{i,1}, \ldots, x_{i,p-1}] \neq [x_{j,1}, \ldots, x_{j,p-1}]$ for any $i \neq j$. If $[x_{i,1}, \ldots, x_{i,p-1}] \neq [0, \ldots, 0]$, then $|\mathbf{B}_i| \geq |\mathbf{S}_i| = 2^{x_{i,p} + x_{i,p+1} + \cdots + x_{i,n}}$. If $[x_{i,1}, \ldots, x_{i,p-1}] = [0, \ldots, 0]$, then $|\mathbf{B}_i| = |\mathbf{S}_i| = 2^{x_{i,p} + x_{i,p+1} + \cdots + x_{i,n}} - 1$ since $[0, \ldots, 0] \notin \mathbf{B}_i$. □

Data Availability

The handwritten digits data supporting this study are from previously reported studies and datasets, which have been cited. The processed data are available in the MNIST database.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] X. Zhang, C. Liu, and C. Y. Suen, "Towards robust pattern recognition: a review," *Proceedings of the IEEE*, vol. 108, no. 6, pp. 894–922, 2020.

[2] S. Theodoridis and K. Koutroumbas, "Pattern recognition," *Features, Feature Vectors, and Classifiers*, Elsevier, Amsterdam, Netherlands, Chapter 1.2, 4th edition, 2010.

[3] L.-J. Cai, S. Lv, and K.-b. Shi, "Application of an improved chi feature selection algorithm," *Discrete Dynamics in Nature and Society*, vol. 2021, Article ID 9963382, 8 pages, 2021.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] Q. Zheng, M. Yang, X. Tian, N. Jiang, and D. Wang, "A full stage data augmentation method in deep convolutional neural network for natural image classification," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 4706576, 11 pages, 2020.

[6] C.-M. Yuan, X.-M. Sun, and Z. Hu, "Speech separation using convolutional neural network and attention mechanism," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 2196893, 10 pages, 2020.

[7] Y. Li, Z. Feng, S. Chen, Z. Zhao, and F. Wang, "Application of the artificial neural network and support vector machines in forest fire prediction in the guangxi autonomous region, China," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 5612650, 14 pages, 2020.

[8] Yu Liang, S. Li, C. Yan, M. Li, and C. Jiang, "Explaining the black-box model: a survey of local interpretation methods for deep neural networks," *Neurocomputing*, vol. 419, pp. 168–182, 2020.

[9] J. Ukita, "Causal importance of low-level feature selectivity for generalization in image recognition," *Neural Networks*, vol. 125, pp. 185–193, 2020.

[10] D. Li, X. Hu, C.-jie Jin, and J. Zhou, "Learning to detect traffic incidents from data based on tree augmented naive bayesian classifiers," *Discrete Dynamics in Nature and Society*, vol. 2017, Article ID 8523495, 9 pages, 2017.

[11] M. Kulin, T. Kazaz, I. Moerman, and E. De Poorter, "End-to-end learning from spectrum data: a deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18484–18501, 2018.

[12] J. Pearl and D. Mackenzie, *The Book of Why, Chapter Introduction, Mind Over Data*, Penguin Books, London, UK, 2018.

[13] L. L. Jacoby and C. M. Kelley, "Unconscious influences of memory for a prior event," *Personality and Social Psychology Bulletin*, vol. 13, no. 3, pp. 314–336, 1987.

[14] D. L. Schacter, "Implicit memory: history and current status," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 13, no. 3, p. 501, 1987.

[15] E. L. Glisky, *Implicit Memory*, Springer, Berlin, Germany, 2018.

[16] I. A. García-León, N. Pereda Beltrán, A. Alanis, and J. S. Moreno, "Intelligent system for the evaluation of implicit memory with semantic emotional stimuli (is-eimses)," in *Agents and Multi-Agent Systems: Technologies and Applications 2021*, G. JezicJ. Chen-Burger et al., Eds., p. 337, Article ID 347, Springer, Berlin, Germany, 2021.

[17] N. J. Cohen, E. Howard, B. S. Deacedo, and S. Corkin, "Different memory systems underlying acquisition of procedural and declarative knowledge," *Annals of the New York Academy of Sciences*, vol. 444, no. 1, pp. 54–71, 1985.

[18] J. Lin, Y. Meng, and W. Lin, "Conditional automaticity: interference effects on the implicit memory retrieval process," *Psychological Research*, vol. 85, no. 1, pp. 223–237, 2021.

[19] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[20] T. L. Floyd, "Digital fundamentals," *The Universal Property of NAND and NOR Gates and 4.9 Karnaugh Map SOP Minimization*, Publishing House of Electronics Industry, Beijing, China, Chapter 5.3, 11th edition, 2017.

[21] D. Attwell and S. B. Laughlin, "An energy budget for signaling in the grey matter of the brain," *Journal of Cerebral Blood Flow & Metabolism*, vol. 21, no. 10, pp. 1133–1145, 2001.

[22] L. Peter, "The cost of cortical computation," *Current Biology*, vol. 13, no. 6, pp. 493–497, 2003.

[23] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[24] U. Lall and A. Sharma, "A nearest neighbor bootstrap for resampling hydrologic time series," *Water Resources Research*, vol. 32, no. 3, pp. 679–693, 1996.

[25] S. E. Taabat, "A novel multicriteria decision-making method based on distance, similarity, and correlation: Dsc topsis," *Mathematical Problems in Engineering*, vol. 2019, Article ID 9125754, 20 pages, 2019.

[26] L. Zhou, Q. Li, G. Huo, and Y. Zhou, "Image classification using biomimetic pattern recognition with convolutional neural networks features," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 3792805, 12 pages, 2017.

[27] Y. Lecun, *MNIST Handwritten Digit Database, Yann Lecun, Corinna Cortes and Chris burges*, BibSonomy, 2013.

[28] H. J. Seo and P. Milanfar, "Training-free, generic object detection using locally adaptive regression kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1688–1704, 2010.

[29] J. Xu, P. Xu, Z. Wei, Xu Ding, and L. Shi, "Dc-nnmn: across components fault diagnosis based on deep few-shot learning," *Shock and Vibration*, vol. 2020, Article ID 3152174, 11 pages, 2020.

[30] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–14, 2017.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 2230805, 14 pages
https://doi.org/10.1155/2021/2230805

Research Article

Decentralized Adaptive Control for Quasi-Consensus in Heterogeneous Nonlinear Multiagent Systems

Jiaju An,[1] Wei Yang,[1] Xiaohui Xu,[2] Tianxiang Chen,[1] Bin Du,[1] Yi Tang,[1] and Quan Xu[1]

[1]*School of Mechanical Engineering, Xihua University, Chengdu 610039, China*
[2]*School of Automobile and Transportation, Xihua University, Chengdu 610039, China*

Correspondence should be addressed to Quan Xu; quanxnjd@sina.com

Received 19 May 2021; Accepted 2 July 2021; Published 14 July 2021

Academic Editor: Guoguang Wen

This paper proposes some novel decentralized adaptive control protocols to settle the quasi-consensus problem of multiagent systems with heterogeneous nonlinear dynamics. Based on local communication with the leader and between the followers, some innovative control protocols are put forward to adapt the control gains and coupling weights simultaneously and to steer the consensus errors to some bounded areas. In particular, two new inequalities are proposed to establish the Lyapunov-based adaptive controller design approach for quasi-consensus. Some quasi-consensus criteria are derived by utilizing the designed controllers, in which the error bound can be modulated on the basis of the adaptive controller parameters. Numerical tests are conducted to show the feasibility of the theoretical derivation. Our findings highlight quasi-consensus in heterogeneous multiagent systems without adding some additional complex nonlinear control terms to cancel the dynamical differences between agents.

1. Introduction

Collaborative control of multiagent systems (MASs) has become a research hotspot of distributed artificial intelligence because of its wide application in intelligent energy, multirobot formation, intelligent transportation, multi-unmanned system collaboration, and other engineering systems [1–5].

Among them, consensus or synchronization is a key common scientific issue of cooperative control of MASs, which has aroused great concern of multidisciplinary scholars. In brief, a core issue is to design appropriate control protocols to reach an agreement between agents. So far, many significant results have been acquired, regarding the consensus patterns, models, control algorithms, etc (please see [6–14] and some other results).

However, the aforesaid works [1–14] on the consensus of MASs mainly concentrate upon the case in which an agent has the same dynamics as all the neighbors or the leader. In some real scenes, the mismatched parameters or dynamical differences among agents may be almost inevitable, which will thus result in heterogeneous (or nonidentical) multiagent systems (HMASs) [15–17]. It is remarkably that, due to heterogeneity, it is even impossible for HMASs to reach the complete consensus just by state feedback control when the coupling weights are constant. Up to now, there are few thorough research studies on complete consensus in HMASs because one has to add some additional complex nonlinear control terms or design compensators to cancel the dynamical differences [18–26]. Another alternative technique to deal with the heterogeneity is to transform HMASs to homogeneous ones [27, 28]. Unfortunately, all of the aforementioned approaches are complex and nonintuitive, which are thus not suitable for engineering applications. Nevertheless, in many practical HMASs, the consensus error may be bounded, even small enough, which is, namely, the so-called quasi-consensus (QC) [29].

Instead, an immediate and natural question, then, is how to design a simple controller to reach QC in HMASs. Similar to it, work to date has considered the quasi-synchronization (QS) in heterogeneous complex dynamical networks (HCDN) [30–34]. For example, in [33, 34], some QS criteria

for fractional HCDN are derived via state feedback control and impulsive control, respectively. By contrast, there are few research studies about QC of HMASs [29, 35–38]. The definition of QC for MASs was first proposed in [38], and then, the definition of QC was further broadened in [29]. After that, the QC of HMASs has been further studied. The QC problem of nonlinear HMASs is studied via sampled-data control in [35]. In [36], sufficient conditions for the QC in switched HMASs are given, considering cooperation and competition interactions simultaneously. Ye and Shao [37] attempted to prove that QC in HMASs under DOS attacks can be realized by impulsive control. It should be noted, however, that, for large scale HCDN or HMASs, the computational complexity and conservativeness of QS or QC conditions in [29–38] impede their applications. In particular, it is difficult, even impossible, to check the LMI-based conditions without adaptive schemes for large-scale HCDN or HMASs. Obviously, it is an interesting and open problem to improve the QC conditions for HMASs by exploring some new control algorithms.

Motivated by the applications of decentralized adaptive control for synchronization in integer-order and fractional-order complex dynamical networks [39–43], this paper aims to design some decentralized adaptive protocols to reach QC in HMASs. Different from some previous studies [37, 44, 45], in our controller, the coupling and feedback values between agents change adaptively. The main contributions in this paper are as follows. First, two new inequalities are proposed to establish the Lyapunov-based adaptive controller design approach for QC in HMASs. Second, some innovative control protocols are introduced to accommodate the control gains and coupling weights adaptively, to steer the consensus errors to some bounded areas. Third, some QC criteria under the designed controllers are derived by the Lyapunov function method and the new inequalities.

2. Preliminaries and Problem Formation

2.1. Graph Theory. To carry out later research, we present some important concepts about graph theory in this section.

$\mathscr{G} = \{\mathscr{V}, \mathscr{E}, \mathscr{A}\}$ is defined as a weighted undirected graph with the network topology of N agents, where $\mathscr{V} = \{0, 1, \ldots, N\}$ and $\mathscr{E} = \{e_{ij}(i, j)\} \subseteq \mathscr{V} \times \mathscr{V}$ are the separate sets of nodes and undirected edges. $\mathscr{A} = (a_{ij})_{N\times N}$ is the weighted adjacency matrix of which the elements are nonnegative. The ith agent and the leader are modeled as the node $i \in \mathscr{V}, \mathscr{V} = \{0, 1, \ldots, N\}$ and node 0, respectively. As a rule, the undirected edge $(i, j) \in \mathscr{E}$ in the weighted undirected graph $\mathscr{G}$ denoted by an ordered pair $(\mathscr{V}_i, \mathscr{V}_j)$ represents that agent i and agent j become a pair of neighbors which can get their information from each other. There is an undirected spanning tree in an undirected graph if one agent exits an undirected path to every other distinct node.

There are two matrices that are considered as the network topology, i.e., the weighted adjacency matrix $\mathscr{A} = (a_{ij})_{N\times N}$ with $a_{ij} = a_{ji} > 0$ if $e_{ij} \in \mathscr{E}$, else $a_{ij} = 0$ if $e_{ij} \notin \mathscr{E}$, and the Laplacian matrix $L = (L_{ij})_{N\times N}$ which is defined as $L_{ii} = \sum_{j=1, j\neq i}^{N} a_{ij} = \deg(i)$ and $L_{ij} = -a_{ij}$, $i \neq j$, noted that all of them are asymmetric for the undirected graph.

Lemma 1 (see [46]). *The Laplacian matrix L is constructed from the undirected network. There are several properties in the following.*

(1) Eigenvalues of L satisfy $0 = \lambda_1(L) < \lambda_2(L) \leq \cdots \leq \lambda_N(L)$ and the smallest positive eigenvalue $\lambda_2(L) = \min_{x^T 1_N = 0, x\neq 0}(x^T Lx/x^T x)$ if and only if the network is connected.

(2) For any vector $\eta = (\eta_1, \eta_2, \ldots, \eta_N)^T \in \mathbb{R}^N$, the equation satisfies

$$\eta^T L\eta = \frac{1}{2}\sum_{i=1}\sum_{j=1} G_{ij}\left(\eta_i - \eta_j\right)^2. \tag{1}$$

2.2. Problem Formation. In this paper, we consider that there are N follower multiagent systems, which can be described by

$$\dot{\omega}_i(t) = A_i\omega_i(t) + B_i f\left(\omega_i(t), t\right) + u_i(t), \tag{2}$$

where $\omega_i(t) \in R^n$ can be regarded as the position vector of the agent $i.f\colon R^n \times R^n \times R^+ \longrightarrow R^n$ represents a continuous nonlinear vector function. A_i, $B_i \in \mathbb{R}^{n\times n}$ represent the system matrices of the agent i, respectively. $u_i(t) \in R^n$ denotes the control protocol to be designed.

The leader agent can be described as follows:

$$\dot{\omega}_0(t) = A\omega_0(t) + Bf\left(\omega_0(t), t\right), \tag{3}$$

where $\omega_0(t) \in R^n$ represents the leader's state vector and A, $B \in \mathbb{R}^{n\times n}$ represent the leader's system matrices, respectively.

For the above heterogeneous leader-follower multiagent system, we give the following assumptions.

Assumption 1. Suppose there is a normal quantity l so that the vector function f for any vector $\lambda, \nu \in R^n$ satisfies

$$\|f(\lambda, t) - f(\nu, t)\| \leq l\|(\lambda - \nu)\|. \tag{4}$$

Lemma 2 (see [47]). *For any vector $x, y \in R^n$, the following holds:*

$$x^T y \leq \frac{1}{2}x^T x + \frac{1}{2}y^T y. \tag{5}$$

Lemma 3. *Any two continuous functions satisfy*

$$\dot{v}(t) + \dot{w}(t) \leq -\gamma v(t), \tag{6}$$

where $\gamma > 0$. Then, there is a $t \geq 0$ so that the following holds:

$$v(t) \leq (v(0) + w(0))e^{-\gamma t} - w(t) + \gamma e^{-\gamma t} * w(t), \quad t \geq 0, \tag{7}$$

where $*$ is the convolution.

Proof. Since $v(t) + \dot{w}(t) \leq -\gamma v(t)$, the existence of $z(t) \geq 0$ makes the following equation:

$$\dot{v}(t) + \dot{w}(t) + z(t) = -\gamma v(t). \tag{8}$$

We calculate the Laplace transform of (8) to obtain

$$sV(s) + sW(s) - (v(0) + w(0)) + Z(s) = -\gamma V(s). \tag{9}$$

Then, we can obtain

$$V(s) = (v(0) + w(0))\frac{1}{s+\gamma} - \frac{s}{s+\gamma}W(s) - \frac{1}{s+\gamma}Z(s). \tag{10}$$

The inverse Laplace transform of (10) can be obtained as

$$v(t) = (v(0) + w(0))e^{-\gamma t} - w(t) + \gamma e^{-\gamma t} * w(t) - e^{-\gamma t} * z(t). \tag{11}$$

Through (10), we can obtain

$$v(t) \leq (v(0) + w(0))e^{-\gamma t} - w(t) + \gamma e^{-\gamma t} * w(t). \tag{12}$$

Lemma 4. *Any two continuous functions satisfy*

$$\dot{v}(t) + \dot{w}(t) \leq -\gamma v(t) + \varepsilon, \tag{13}$$

where $\gamma > 0$ and $\varepsilon > 0$. Then, there is a $t \geq 0$ so that the following formula holds:

$$v(t) < (v(0) + w(0))e^{-\gamma t} - w(t) + \gamma e^{-\gamma t} * w(t) + \frac{\varepsilon}{\gamma}, \quad t \geq 0. \tag{14}$$

Proof. Since $\dot{v}(t) + \dot{w}(t) \leq -\gamma v(t) + \varepsilon$, then $(v(t) - (\varepsilon/\gamma))^{\cdot} + \dot{w}(t) \leq -\gamma(v(t) - (\varepsilon/\gamma))$.

By Lemma 3, we can obtain

$$v(t) - \frac{\varepsilon}{\gamma} \leq \left(v(0) - \frac{\varepsilon}{\gamma} + w(0)\right)e^{-\gamma t} - w(t) + \gamma e^{-\gamma t} * w(t). \tag{15}$$

We can further obtain

$$v(t) \leq (v(0) + w(0))e^{-\gamma t} - w(t) + \gamma e^{-\gamma t} * w(t) + \frac{\varepsilon}{\gamma}. \tag{16}$$

Lemma 5 (see [47]). *If A, B, C, and D represent four different matrices, respectively, and the matrix products AC and B D makes sense, the Kronecker product $\otimes$ satisfies*

$$\begin{aligned} &(1)\ A \otimes (B + C) = A \otimes B + A \otimes C, \\ &(2)\ (A \otimes B)(C \otimes D) = (AC) \otimes (B\ D). \end{aligned} \tag{17}$$

Definition 1 (see [29]). The leader-follower HMASs are decided to reach QC if

$$\lim_{t \longrightarrow \infty} \left\| \omega_i(t) - \omega_0(t) \right\| \leq \xi, \quad i = 1, 2, \ldots, N, \tag{18}$$

where ξ is a nonnegative constant.

Assumption 2. The network topology between agents is undirected and connected, and each agent can acquire the status information of the agent that has a connection relationship with it and the leader agent at any time.

2.3. Our Controller. To obtain QC between HMASs (2) and (3), we design the control input for all follower agents as

$$u_i(t) = -c\sum_{j=1}^{N} L_{ij}(t)\omega_j(t) - r_i(t)\left(\omega_i(t) - \omega_0(t)\right), \tag{19}$$

where c is a positive constant.

The adaptive law for the control gains is described as

$$\dot{r}_i(t) = \mu\left(\omega_i(t) - \omega_j(t)\right)^T\left(\omega_i(t) - \omega_0(t)\right), \tag{20}$$

where μ is a positive constant to be selected.

The adaptive law for the coupling weights is described as

$$\begin{aligned} &\dot{L}_{ij}(t) = -\alpha_{ij}\left(\omega_i(t) - \omega_j(t)\right)^T\left(\omega_i(t) - \omega_j(t)\right), \\ &L_{ij}(0) = L_{ji}(0) > 0, \quad (i, j) \in \mathscr{E}, \end{aligned} \tag{21}$$

where $\alpha_{ij} = \alpha_{ji}$ are the positive constants to be selected.

Remark 1. It should be noted that, in controller (19), adaptive laws (20) and (21), the coupling weights $L_{ij}(t)$ and the control gains $r_i(t)$ are adjusted adaptively based on local communication with the leader and between the followers. Combining adaptation of the coupling weights and control gains, adaptive law (20) ensures the QC of the follower agents, while adaptive law (21) drives the follower agents to the leader agent.

Let the QC error vector be $e_i(t) = \omega_i(t) - \omega_0(t)$. Then, the error model with the controller iscom

$$\begin{aligned} \dot{e}_i(t) = {} & A_i e_i(t) + B_i \widetilde{f}\left(e_i(t), t\right) + h_i\left(\omega_0(t), t\right) \\ & - c\sum_{j=1}^{N} L_{ij}(t)e_j(t) - r_i(t)e_i(t), \end{aligned} \tag{22}$$

where $\widetilde{f}(e_i(t), t) = f(\omega_i(t), t) - f(\omega_0(t), t)$ and $h_i(\omega_0(t), t) = (A_i - A)\omega_0(t) + (B_i - B)f(\omega_0(t), t)$.

$h_i(\omega_0(t), t)$ represents the difference between different agents. It can be obtained by Assumption 1:

$$\begin{aligned} \left\| h_i\left(\omega_0(t), t\right) \right\| &= \left\| (A_i - A)\omega_0(t) + (B_i - B)f\left(\omega_0(t), t\right) \right\| \\ &\leq \left\| A_i - A \right\| \left\| \omega_0(t) \right\| + \left\| B_i - B \right\| \left\| f\left(\omega_0(t), t\right) \right\| \\ &\leq \left\| A_i - A \right\| \left\| \omega_0(t) \right\| + \left\| B_i - B \right\| l_{\max} \left\| \omega_0(t) \right\|. \end{aligned} \tag{23}$$

Since $\omega_0(t)$ is bounded, it can be obtained that $h_i(\omega_0(t), t)$ is bounded.

3. Main Results

In this section, we present some theorems for achieving QC in HMASs.

3.1. Adaptive Control Protocol

Theorem 1. *It is assumed that $f(\omega_i(t),t)$ satisfies Assumption 1 and the follower system (2) satisfies Assumption 2. Under the action of controller (19) and adaptive laws (20) and (21), HMASs (2) and (3) can achieve QC.*

The Lyapunov function we constructed is

$$V_1(t) = \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)e_i(t) + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\frac{c}{4\alpha_{ij}}\left(L_{ij}(t)+k_{ij}\right)^2 + \sum_{i=1}^{N}\frac{1}{2\mu}\left(r_i(t)-d_i^*\right)^2, \tag{24}$$

where $k_{ij} = k_{ji}\,(i \neq j)$ is a nonnegative constant if and only if $L_{ij}(t) = 0$ is $k_{ij} = 0$. d_i^* is the normal constant waiting for the value.

We take the derivative of $V_1(t)$ along (22) together with controller (19) and adaptive laws (20) and (21), and we can obtain

$$\begin{aligned}
\dot{V}_1(t) &= \sum_{i=1}^{N} e_i^T(t)\dot{e}_i(t) + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\frac{c}{2\alpha_{ij}}\left(L_{ij}(t)+k_{ij}\right)\dot{L}_{ij}(t) + \frac{1}{\mu}\sum_{i=1}^{N}\left(r_i(t)-d_i^*\right)\dot{r}_i(t)\\
&= \sum_{i=1}^{N} e_i^T(t)\left(A_ie_i(t) + B_i\tilde{f}(e_i(t),t) + h_i(\omega_0(t),t) - c\sum_{j=1}^{N}L_{ij}(t)e_j(t) - r_i(t)e_i(t)\right)\\
&\quad - \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\frac{c}{2}\left(L_{ij}(t)+k_{ij}\right)\left(\omega_i(t)-\omega_j(t)\right)^T\left(\omega_i(t)-\omega_j(t)\right)\\
&\quad + \sum_{i=1}^{N}\left(r_i(t)-d_i^*\right)e_i^T(t)e_i(t)\\
&= \sum_{i=1}^{N} e_i^T(t)A_ie_i(t) + \sum_{i=1}^{N} e_i^T(t)B_i\tilde{f}(e_i(t),t) + \sum_{i=1}^{N} e_i^T(t)h_i(\omega_0(t),t)\\
&\quad - \sum_{i=1}^{N} e_i^T(t)c\sum_{j=1}^{N}L_{ij}(t)e_j(t) - \sum_{i=1}^{N} d_i^*e_i^T(t)e_i(t)\\
&\quad - \frac{c}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\left(L_{ij}(t)+k_{ij}\right)\left(\omega_i(t)-\omega_j(t)\right)^T\left(\omega_i(t)-\omega_j(t)\right).
\end{aligned} \tag{25}$$

Through Assumption 1 and Lemma 2, one has

$$\begin{aligned}
\sum_{i=1}^{N} e_i^T(t)B_i\tilde{f}(e_i(t),t) &= \sum_{i=1}^{N} e_i^T(t)B_i\left(f(\omega_i(t),t) - f(\omega_0(t),t)\right)\\
&\leq \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)B_iB_i^Te_i(t) + \frac{1}{2}\sum_{i=1}^{N}\left\|f(\omega_i(t),t) - f(\omega_0(t),t)\right\|^2\\
&\leq \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)\left(B_iB_i^T + l^2I_n\right)e_i(t),
\end{aligned} \tag{26}$$

$$\sum_{i=1}^{N} e_i^T(t)h_i(\omega_0(t),t) \leq \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)e_i(t) + \frac{1}{2}\sum_{i=1}^{N}\left\|h_i(\omega_0(t),t)\right\|^2. \tag{27}$$

From (25)–(27), we can obtain

$$\begin{aligned}\dot{V}_1(t) \le & \sum_{i=1}^{N} e_i^T(t) A_i e_i(t) + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t) e_i(t) + \frac{1}{2}\sum_{i=1}^{N} \left\| h_i(\omega_0(t), t)\right\|^2 \\ & - \sum_{i=1}^{N} e_i^T(t) c \sum_{j=1}^{N} L_{ij}(t) e_j(t) + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)\left(B_i B_i^T + l^2 I_n\right) e_i(t) - \sum_{i=1}^{N} d_i^* e_i^T(t) e_i(t) \\ & - \frac{c}{2}\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} \left(L_{ij}(t) + k_{ij}\right)\left(\omega_i(t) - \omega_j(t)\right)^T\left(\omega_i(t) - \omega_j(t)\right).\end{aligned} \tag{28}$$

We can define the Laplacian matrix $\Omega = (\tau_{ij})_{N\times N}$, where $\tau_{ij} = k_{ij}, i \neq j$ and $\tau_{ii} = -\sum_{\substack{j=1\\ j\neq i}}^{N} \tau_{ij}$. By Lemma 1, we can obtain

$$\begin{aligned}& \frac{c}{2}\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} \left(L_{ij}(t) + k_{ij}\right)\left(e_i(t) - e_j(t)\right)^T\left(e_i(t) - e_j(t)\right) \\ & = -c\sum_{i=1}^{N}\sum_{j=1}^{N} L_{ij}(t) e_i^T(t) e_j(t) + c\sum_{i=1}^{N}\sum_{j=1}^{N} \tau_{ij} e_i^T(t) e_j(t).\end{aligned} \tag{29}$$

Combining with (28) and (29), we have

$$\begin{aligned}\dot{V}_1(t) \le & \sum_{i=1}^{N} e_i^T(t) A_i e_i(t) + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t) e_i(t) + \frac{1}{2}\sum_{i=1}^{N} \left\| h_i(\omega_0(t), t)\right\|^2 \\ & - \sum_{i=1}^{N} e_i^T(t) c \sum_{i=1}^{N} L_{ij}(t) e_j(t) + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)\left(B_i B_i^T + l^2 I_n\right) e_i(t) \\ & + c\sum_{i=1}^{N}\sum_{i=1}^{N} L_{ij}(t) e_i^T(t) e_j(t) - c\sum_{i=1}^{N}\sum_{i=1}^{N} \tau_{ij} e_i^T(t) e_j(t) - \sum_{i=1}^{N} d_i^* e_i^T(t) e_i(t) \\ = & \sum_{i=1}^{N} e_i^T(t) A_i e_i(t) + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)\left(B_i B_i^T + \left(l^2 + 1\right) I_n - 2 d_i^* \cdot I_n\right) e_i(t) \\ & + \frac{1}{2}\sum_{i=1}^{N} \left\| h_i(\omega_0(t), t)\right\|^2 - c\sum_{i=1}^{N}\sum_{i=1}^{N} \tau_{ij} e_i^T(t) e_j(t).\end{aligned} \tag{30}$$

Then, there is a unitary matrix $U = (u_1, \ldots, u_N)$ so that $U^T \Omega U = \Delta$, $m(t) = (U^T \otimes I_n) e(t)$,

$$\begin{aligned}\dot{V}_1(t) \le & e^T(t)\left[A + \frac{1}{2}\left(BB^T + I_N \otimes \left(l^2 + 1\right) I_n - 2\left(D^* \otimes I_n\right)\right) - c\left(\Omega \otimes I_n\right)\right] e(t) + \frac{1}{2}\left\| h(\omega_0(t), t)\right\|^2 \\ = & e^T(t)\left[A + \frac{1}{2}\left(BB^T + I_N \otimes \left(l^2 + 1\right) I_n - 2\left(D^* \otimes I_n\right)\right)\right] \\ & - c\left(U \otimes I_n\right)\left(\Delta \otimes I_n\right)\left(U^T \otimes I_n\right)\Big)\Big] e(t) + \frac{1}{2}\left\| h(\omega_0(t), t)\right\|^2 \\ = & e^T(t)\left[A + \frac{1}{2}\left(BB^T + I_N \otimes \left(l^2 + 1\right) I_n - 2\left(D^* \otimes I_n\right)\right)\right] e(t) - c m^T(t)\left(\Delta \otimes I_n\right) m(t) \\ & + \frac{1}{2}\left\| h(\omega_0(t), t)\right\|^2,\end{aligned} \tag{31}$$

where $D^* = \mathrm{diag}(d_1^*, d_2^*, \ldots, d_N^*)$, $B = \mathrm{diag}(B_1, B_2, \ldots, B_N)$, and $A = \mathrm{diag}(A_1, A_2, \ldots, A_N)$.

Since I_n is positive definite, by Lemma 1, we can get the following:

$$m^T(t)(\Delta \otimes I_n)m(t) \geq \lambda_2(\Omega)m^T(t)(I_N \otimes I_n)m(t). \tag{32}$$

Then, we can obtain

$$\begin{aligned}
\dot{V}_1(t) &\leq e^T(t)\left[A + \frac{1}{2}\left(BB^T + I_N \otimes \left(l^2+1\right)I_n - 2\left(D^* \otimes I_n\right)\right)\right]e(t) + \frac{1}{2}\left\|h\left(\omega_0(t), t\right)\right\|^2 \\
&\quad - c\lambda_2(\Omega)m^T(t)\left(I_N \otimes I_n\right)m(t) \\
&= e^T(t)\left[A + \frac{1}{2}\left(BB^T + I_N \otimes \left(l^2+1\right)I_n - 2\left(D^* \otimes I_n\right)\right)\right]e(t) + \frac{1}{2}\left\|h\left(\omega_0(t), t\right)\right\|^2 \\
&\quad - c\lambda_2(\Omega)e^T(t)\left(P \otimes I_n\right)\left(I_N \otimes I_n\right)\left(P^T \otimes I_n\right)e(t) \\
&= e^T(t)\left[A + \frac{1}{2}\left(BB^T + I_N \otimes \left(l^2+1\right)I_n - 2\left(D^* \otimes I_n\right)\right) - c\lambda_2(\Omega)\left(I_N \otimes I_n\right)\right]e(t) \\
&\quad + \frac{1}{2}\left\|h\left(\omega_0(t), t\right)\right\|^2.
\end{aligned} \tag{33}$$

We can choose large enough k_{ij} and d_i^* so that

$$\lambda_{\max}\left(A + \frac{1}{2}\left(BB^T + I_N \otimes \left(l^2+1\right)I_n - 2\left(D^* \otimes I_n\right)\right) - c\lambda_2(\Omega)\left(I_N \otimes I_n\right)\right) + \theta \leq 0, \tag{34}$$

where $\theta > 0$ is a positive constant.

$$\dot{V}_1(t) \leq -\theta e^T(t)e(t) + \frac{1}{2}\left\|h\left(\omega_0(t), t\right)\right\|^2, \tag{35}$$

where $e(t) = (e_1^T, e_2^T, e_3^T, \ldots, e_N^T)^T$.

By Lemma 4, it yields

$$\begin{aligned}
e^T(t)e(t) \leq &\left(\sum_{i=1}^{N} e_i^T(0)e_i(0) + \sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} \frac{c}{2\alpha_{ij}}\left(L_{ij}(0) - k_{ij}\right)^2\right) \\
&+ \sum_{i=1}^{N} \frac{1}{\mu}\left(r_i(0) - d_i^*\right)^2\Bigg)e^{-2\theta t} + 2\theta e^{-2\theta t} * \left(\sum\sum_{i=1}^{N} \frac{1}{\mu}\left(r_i(0) - d_i^*\right)^2\right. \\
&\left. + \sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} \frac{c}{2\alpha_{ij}}\left(L_{ij}(0) - k_{ij}\right)^2\right) + \frac{\left\|h\left(\omega_0(t), t\right)\right\|^2}{2\theta}.
\end{aligned} \tag{36}$$

The error will eventually converge to finite region $\Xi = \left\{e(t) \in R^n \middle| \|(t)\| \leq \sqrt{\left(\|h(\omega_0(t), t)\|^2/2\theta\right)}\right\}$ as $t \longrightarrow +\infty$. This completes the proof.

3.2. Adaptive Pinning Control Protocol. In the previous section, an adaptive controller (19) and an adaptive laws (20) and (21) for all follower agents are designed. However, it is neither realistic nor economical to control all follower agents in engineering. In view of this, the adaptive pinning control schemes are considered, where a fraction of the control gains and the coupling weights are adapted.

Suppose that $\tilde{\varepsilon}$ is a subset of $\mathscr{E}$ and HMASs (2) and (3) are connected by the undirected edge. $\tilde{\varepsilon}$ is connected. The pinning adaptive protocol is written as follows:

$$u_i(t) = -c\sum_{j=1}^{N} L_{ij}(t)\omega_j(t) - \vartheta_i r_i(t)\left(\omega_i(t) - \omega_0(t)\right), \tag{37}$$

where $\vartheta_i = \begin{cases} 1, & \text{for } i = 1, 2, \ldots, N_s, \\ 0, & \text{for } i = N_s + 1, \ldots, N \end{cases}$.

The adaptive law for $r_i(t)$ is the same as (20).

The adaptive law for $L_{ij}(t)$ is described as

$$\dot{L}_{ij}(t) = -\alpha_{ij}\left(\omega_i(t) - \omega_j(t)\right)^T\left(\omega_i(t) - \omega_j(t)\right), \tag{38}$$

where $(i, j) \in \tilde{\varepsilon}$ and $L_{ij}(0) = L_{ji}(0) > 0$.

Theorem 2. *Suppose that Assumption 1 is valid and the follower system (2) satisfies Assumption 2. The HMASs (2) and (3) can achieve QC under the pinning adaptive protocol combined with (20), (21), and (37).*

Proof. Take into account the Lyapunov function candidate:

$$V_2(t) = \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)e_i(t) + \sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N}\frac{c}{4\alpha_{ij}}\left(L_{ij}(t) + k_{ij}\right)^2 + \sum_{i=1}^{N_s}\frac{1}{2\mu}\left(r_i(t) - d_i^*\right)^2, \tag{39}$$

where $c_{ij} = c_{ji}\,(i \neq j)$ is a nonnegative quantity if and only if $L_{ij}(t) = 0$ and $k_{ij} = 0$. d_i^* is the normal constant waiting for the value if $i = 1, 2, \ldots, N_s, N_s \geq 1$, or $d_i^* = 0$.

The derivative of $V_2(t)$ along (22), controller (37), and the decentralized adaptive pinning laws (20) and (21) gives

$$\begin{aligned}
\dot{V}_2(t) &= \sum_{i=1}^{N} e_i^T(t)\dot{e}_i(t) + \sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N}\frac{c}{2\alpha_{ij}}\left(L_{ij}(t) + k_{ij}\right)\dot{L}_{ij}(t) + \sum_{i=1}^{N_s}\frac{1}{\mu}\left(r_i(t) - d_i^*\right)\dot{d}_i(t) \\
&\leq \sum_{i=1}^{N} e_i^T(t)A_i e_i(t) + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)\left(B_i B_i^T + \left(l^2 + 1\right)I_n - 2d_i^* \otimes I_n\right)e_i(t) \\
&\quad + \frac{1}{2}\sum_{i=1}^{N}\left\|h_i(\omega_0(t), t)\right\|^2 - c\sum_{i=1}^{N}\sum_{j=1}^{N}\tau_{ij}e_i^T(t)e_j(t) \\
&\leq e^T(t)\left[A + \frac{1}{2}\left(BB^T + I_N \otimes \left(l^2 + 1\right)I_n - 2\left(\tilde{D}^* \otimes I_n\right)\right) - c\lambda_2(\Omega)\left(I_N \otimes I_n\right)\right]e(t) \\
&\quad + \frac{1}{2}\left\|h(\omega_0(t), t)\right\|^2,
\end{aligned} \tag{40}$$

where $\tilde{D}^* = \mathrm{diag}(d_1^*, d_2^*, \ldots, d_{N_s}^*, 0, \ldots, 0)$.

The proof can be completed by using the similar analysis method in Theorem 1.

Theorem 3. *Suppose that Assumptions 1 and 2 hold. The HMASs (2) and (3) can achieve QC under the pinning adaptive protocol combined with (19), (20), and (38).*

Proof. Construct the Lyapunov functional as

$$V_3(t) = \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)e_i(t) + \sum_{i=1}^{N}\sum_{(i,j)\varepsilon\tilde{\varepsilon}}\frac{c}{4\alpha_{ij}}\left(L_{ij}(t) + \tilde{k}_{ij}\right)^2 + \sum_{i=1}^{N}\frac{1}{2\mu}\left(r_i(t) - d_i^*\right)^2, \tag{41}$$

where $\tilde{k}_{ij} = \tilde{k}_{ji} > 0$, $(i, j)\varepsilon\tilde{\varepsilon}$, and $\tilde{k}_{ij} = 0\,(i \neq j)$, else.

Let $\tilde{K} = (\tilde{k}_{ij})_{N\times N}, \tilde{k}_{ii} = -\sum_{i=1, i\neq j}^{N}\tilde{k}_{ij}$; then,

$$G_{ij} = \begin{cases} L_{ij}(0), & (i, j) \in E - \tilde{\varepsilon}, \\ -\sum_{j=1, j\neq i}^{N} L_{ij}(0), & i = j, \\ 0, & \text{other}. \end{cases} \tag{42}$$

After the derivative of $V_3(t)$ along (22), controller (19), and the decentralized adaptive pinning laws (20) and (38), the following holds:

$$\begin{aligned}
\dot{V}_3(t) &= \sum_{i=1}^{N} e_i^T(t)\dot{e}_i(t) + \sum_{i=1}^{N}\sum_{(i,j)\varepsilon\varepsilon} \frac{c}{2\alpha_{ij}}\left(L_{ij}(t)+\tilde{k}_{ij}\right)\dot{L}_{ij}(t) + \frac{1}{\mu}\sum_{i=1}^{N}\left(r_i(t)-d_i^*\right)\dot{d}_i(t) \\
&= \sum_{i=1}^{N} e_i^T(t)A_i e_i(t) + \sum_{i=1}^{N} e_i^T(t)B_i\tilde{f}\left(e_i(t),t\right) + \sum_{i=1}^{N} e_i^T(t)h_i\left(\omega_0(t),t\right) \\
&\quad - \sum_{i=1}^{N} e_i^T(t)c\sum_{i=1}^{N} L_{ij}(t)e_j(t) - \sum_{i=1}^{N} d_i^* e_i^T(t)e_i(t) \\
&\quad - \frac{c}{2}\sum_{i=1}^{N}\sum_{(i,j)\varepsilon\varepsilon}\left(L_{ij}(t)+\tilde{k}_{ij}\right)\left(\omega_i(t)-\omega_j(t)\right)^T\left(\omega_i(t)-\omega_j(t)\right) \\
&\le \sum_{i=1}^{N} e_i^T(t)A_i e_i(t) + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)e_i(t) + \frac{1}{2}\sum_{i=1}^{N}\left\|h_i\left(\omega_0(t),t\right)\right\|^2 \\
&\quad + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)\left(B_iB_i^T + l^2 I_n\right)e_i(t) - c\sum_{i=1}^{N}\sum_{j=1}^{N} G_{ij}(t)e_i^T(t)e_j(t) \\
&\quad - c\sum_{i=1}^{N}\sum_{j=1}^{N}\tilde{\tau}_{ij}e_i^T(t)e_j(t) - \sum_{i=1}^{N} d_i^* e_i^T(t)e_i(t) \\
&= e^T(t)\left[A + c\left(G\otimes I_n\right) + \frac{1}{2}\left(I_N\otimes\left(BB^T + \left(l^2+1\right)I_n\right)\right)\right] \\
&\quad \left(D^*\otimes I_n\right) - c\lambda_2\left(\Omega^*\right)\left(I_N\otimes I_n\right)]e(t) + \frac{1}{2}\left\|h\left(\omega_0(t),t\right)\right\|^2,
\end{aligned} \tag{43}$$

where $G = (G_{ij})_{N\times N}$, $\Omega^* = (\tilde{\tau}_{ij})_{N\times N}$, $\tilde{\tau}_{ij} = -\tilde{k}_{ij}, i \neq j$, and $\tilde{\tau}_{ii} = -\sum_{\substack{j=1\\ j\neq i}}^{N}\tilde{\tau}_{ij}$.

The following proof is similar to the previous derivation in Theorem 1; thus, we will omit this part here.

Theorem 4. *Suppose that Assumptions 1 and 2 hold. The HMASs (2) and (3) can achieve QC under the pinning adaptive protocol combined with (20), (37), and (38).*

Proof. The Lyapunov functional is considered as follows:

$$V_4(t) = \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)e_i(t) + \sum_{i=1}^{N}\sum_{(i,j)\varepsilon\tilde{\varepsilon}}\frac{c}{4\alpha_{ij}}\left(L_{ij}(t)+\tilde{k}_{ij}\right)^2 + \sum_{i=1}^{N_s}\frac{1}{2\mu}\left(r_i(t)-d_i^*\right)^2. \tag{44}$$

The derivative of $V_4(t)$ along the error system (22), controller (37), and decentralized adaptive pinning laws (20) and (38) gives

$$\begin{aligned}\dot{V}_4(t) &= \sum_{i=1}^{N} e_i^T(t)\dot{e}_i(t) + \sum_{i=1}^{N}\sum_{(i,j)\varepsilon\varepsilon}^{N} \frac{c}{2\alpha_{ij}}\left(L_{ij}(t)+\tilde{k}_{ij}\right)\dot{L}_{ij}(t) + \frac{1}{\mu}\sum_{i=1}^{N_s}\left(r_i(t)-d_i^*\right)\dot{d}_i(t)\\
&\le \sum_{i=1}^{N} e_i^T(t)A_i e_i(t) + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)e_i(t) + \frac{1}{2}\sum_{i=1}^{N}\left\|h_i(\omega_0(t),t)\right\|^2\\
&\quad + \frac{1}{2}\sum_{i=1}^{N} e_i^T(t)\left(B_iB_i^T + l^2 I_n\right)e_i(t) + c\sum_{i=1}^{N}\sum_{j=1}^{N} G_{ij}(t)e_i^T(t)e_j(t)\\
&\quad - c\sum_{i=1}^{N}\sum_{j=1}^{N}\tilde{\tau}_{ij}e_i^T(t)e_j(t) - \sum_{i=1}^{N_s} d_i^* e_i^T(t)e_i(t)\\
&= e^T(t)\left[A + c\left(G\otimes I_n\right) + \frac{1}{2}\left(I_N\otimes\left(BB^T + \left(l^2+1\right)I_n\right)\right) - \left(\tilde{D}^*\otimes I_n\right)\right]\\
&\quad - c\lambda_2\left(\Omega^*\right)\left(I_N\otimes I_n\right)]e(t) + \frac{1}{2}\left\|h(\omega_0(t),t)\right\|^2,\end{aligned} \tag{45}$$

where $G = (G_{ij})_{N\times N}$, $\Omega^* = (\tilde{\tau}_{ij})_{N\times N}$, $\tilde{\tau}_{ij} = -\tilde{k}_{ij}, i \ne j$, $\tilde{\tau}_{ii} = -\sum_{\substack{j=1\\ j\ne i}}^{N} \tilde{\tau}_{ij}$, and $\tilde{D}^* = \mathrm{diag}(d_1^*, d_2^*, \ldots, d_{N_s}^*, 0, \ldots, 0)$.

The rest of the proof is similar to Theorem 1. To save space, it is thus omitted here.

Remark 2. In Theorems 2–4, some sufficient conditions are given to realize QC of HMASs (2) and (3) by using the adaptive pinning control schemes. Actually, if the follower agents (3) are connected, one can randomly choose a small fraction of coupling weights and/or the control gains to adapt. In particular, it is possible to obtain the QC by pinning one follower agent.

4. Numerical Examples

In this section, we will confirm the theory proposed in the paper by using digital simulation experiments.

We use five following agents and one leader agent, and the initial communication Laplacian matrix for follower agents is

$$L = \begin{pmatrix} 1.5 & -0.3 & -0.3 & -0.4 & -0.5\\ -0.3 & 0.9 & -0.6 & 0 & 0\\ -0.3 & -0.6 & 0.9 & 0 & 0\\ -0.4 & 0 & 0 & 0.7 & -0.3\\ -0.5 & 0 & 0 & -0.3 & 0.8 \end{pmatrix}. \tag{46}$$

The leader's system matrices can be described as $A = \begin{pmatrix} -2.5 & 10 & 0\\ 1 & -1 & 1\\ 0 & -18 & 0\end{pmatrix}$ and $B = \begin{pmatrix} 35/6 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}$, and the nonlinear function of the entire system can be assumed to be $f(t,x) = \begin{pmatrix} 0.5\left(|x_1(t)+1| - |x_1(t)-1|\right)\\ 0\\ 0\end{pmatrix}$. The following system matrices are, respectively, assumed to be $A_i = \begin{pmatrix} -2.5+0.1*i & 10+0.2*i & 0\\ 1+0.2*i & -1+0.2*i & 1+0.2*i\\ 0 & -18+0.2*i & 0\end{pmatrix}$ and $B_i = \begin{pmatrix} 35/6 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}$ $(i = 1, 2, \ldots, 5)$. We arbitrarily choose the initial value as $x_0(0) = (3.3, 0.66, 0.1)^T$ and $x_i(0) = (11.5 + 1.8*i, 4.2 + 1.1*i, 4.9 + 1.5*i)^T$ $(i = 1, 2, \ldots, 5)$. Figure 1 shows the change of $\|e_i(t)\|_2 (i = 1, 2, \ldots, 5)$ for HMASs without the controller.

4.1. Example 1. In this case, the parameters of controller (19) and adaptive laws (20) and (21) are chosen according to Theorem 1. All agent system parameters are in accordance with A_i and B_i. We choose $c = 2.6$, $\mu = 0.05$, $\alpha_{12} = \alpha_{21} = 2.7$, $\alpha_{13} = \alpha_{31} = 2.4$, $\alpha_{14} = \alpha_{41} = 2.1$, $\alpha_{15} = \alpha_{51} = 1.8$, $\alpha_{23} = \alpha_{32} = 2.1$, and $\alpha_{45} = \alpha_{54} = 2.2$. We arbitrarily choose the initial value as $r(0) = (2.3, 2.1, 1.8, 1.8, 2.1)^T$, $x_0(0) = (3.3, 0.66, 0.1)^T$, and $x_i(0) = (11.5 + 1.8*i, 4.2 + 1.1*i, 4.9 + 1.5*i)^T$ $(i = 1, 2, \ldots, 5)$. Figure 2 shows

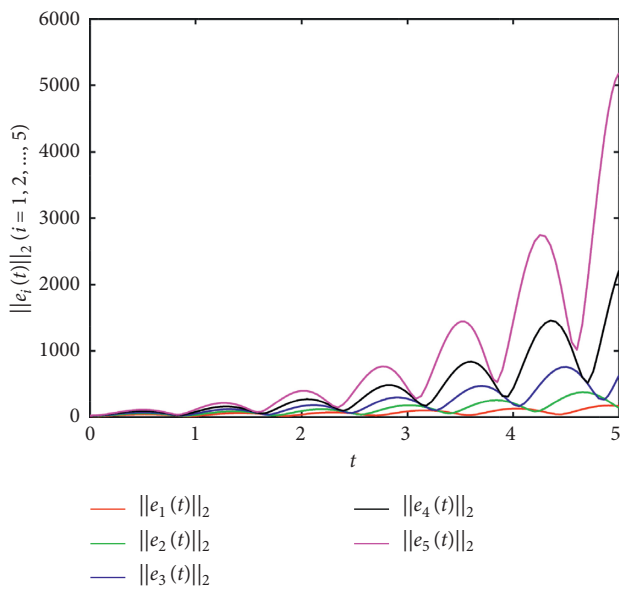


Figure 1: The changes of $\|e_i(t)\|_2 (i = 1, 2, \ldots, 5)$ for the system without thw controller.

the change in the state of the system under the increment controller. It visibly shows that, under the action of the adaptive controller and adaptive law, the error of the leader and follower finally converges to a finite region.

4.2. Example 2. In this example, we only use the coupling between followers and leaders 1 and 2 to achieve QC of the entire system. All agent system parameters are in accordance with A_i and B_i. We choose $c = 2.6$, $\mu = 0.01$, $\alpha_{12} = \alpha_{21} = 8.1$, $\alpha_{13} = \alpha_{31} = 7.2$, $\alpha_{14} = \alpha_{41} = 6.3$, $\alpha_{15} = \alpha_{51} = 5.4$, $\alpha_{23} = \alpha_{32} = 6.3$, and $\alpha_{45} = \alpha_{54} = 7.2$. We arbitrarily choose the 1 and 2 agents, and their initial values are selected as $r(0) = (2.3, 2.1)^T$. The initial value of each agent is selected as $x_0(0) = (3.3, 0.66, 0.1)^T$ and $x_i(0) = (-6 - 1.6 * i, 5 + 1.1 * i, 9.4 + 2 * i)^T (i = 1, 2, \ldots, 5)$. Figure 3 shows $e_i(t)_2 (i = 1, 2, \ldots, 5)$ eventually tends to a finite region.

4.3. Example 3. In this example, we use a control strategy that pinning the coupling between followers to QC of HMASs (2) and (3). All agent system parameters are in accordance with A_i and B_i. We choose $c = 1.2$. The parameters in the adaptive law are $\mu = 0.01$, $\alpha_{14} = \alpha_{41} = 7.2$, $\alpha_{15} = \alpha_{51} = 6.3$, and $\alpha_{23} = \alpha_{32} = 5.4$. We arbitrarily choose the initial value as $r(0) = (2.1, 2.1, 2, 1.9, 2.1)^T$, $x_0(0) = (3.3, 0.66, 0.1)^T$ and $x_i(0) = (-5.3 - 0.5 * i, 3 + 1.6 * i, 2.7 + 1.4 * i)^T (i = 1, 2, \ldots, 5)$. From Figure 4, we can distinctly see that, under the control strategy designed by Theorem 3, $e_i(t)_2 (i = 1, 2, \ldots, 5)$ of HMASs (2) and (3) are concentrated in a limited area.

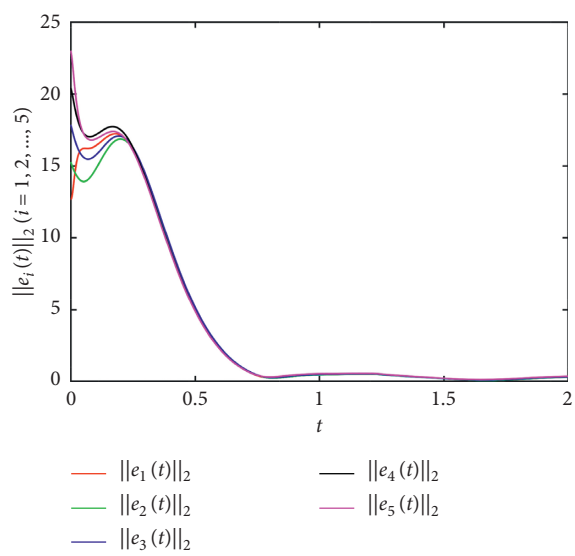


Figure 2: The changes of $\|e_i(t)\|_2 (i = 1, 2, \ldots, 5)$ for the system under the control strategy designed by Theorem 1.

4.4. Example 4. In this case, by controlling the coupling between part of the follower agent and the leader and the coupling between the follower agents, we realize the QC of the whole system. All agent system parameters are in accordance with A_i and B_i. We choose $c = 3$. The parameters in the adaptive law are $\mu = 0.01$, $\alpha_{14} = \alpha_{41} = 4.9$, $\alpha_{15} = \alpha_{51} = 4.2$, and $\alpha_{23} = \alpha_{32} = 4.9$. We arbitrarily choose the 1, 2, and 3 agents, and their initial values are selected as $r(0) = (1.8, 2.1, 1.8)^T$. The initial value of each agent is selected as $x_0(0) = (3.3, 0.66, 0.1)^T$ and $x_1(0) = x_2(0) = x_3(0) = x_4(0) = x_5(0) = (12.3, 5, 5.1)^T$.

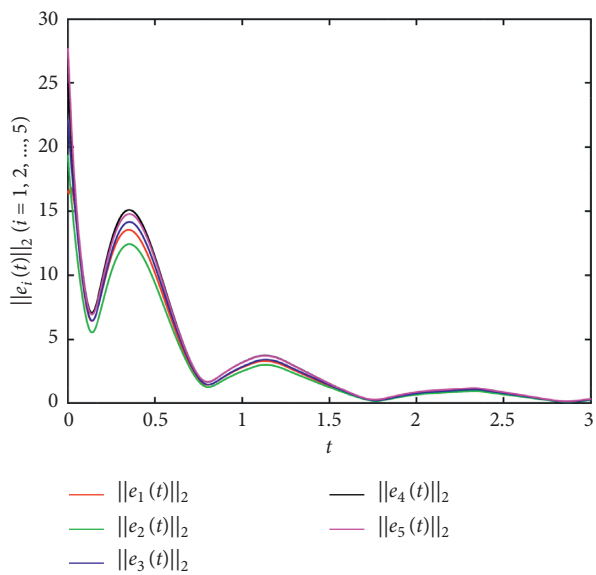


FIGURE 3: The changes of $\|e_i(t)\|_2 (i = 1, 2, \ldots, 5)$ for the system under the control strategy designed by Theorem 2.

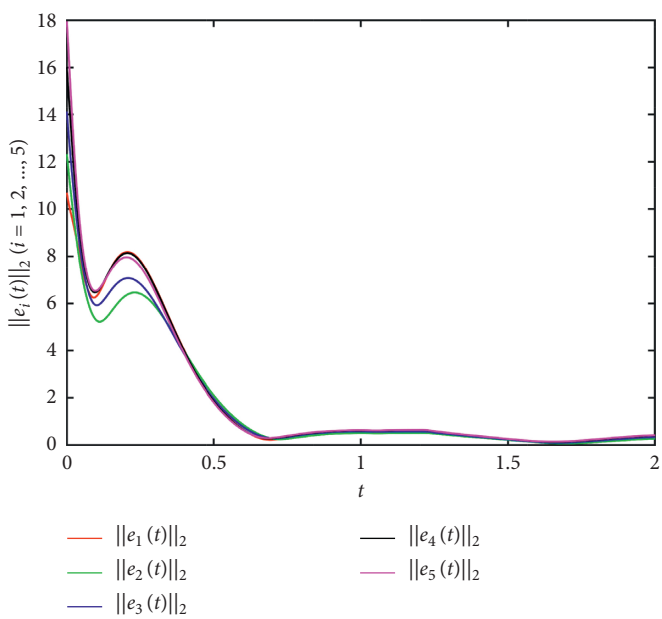


FIGURE 4: The changes of $\|e_i(t)\|_2 (i = 1, 2, \ldots, 5)$ for the system under the control strategy designed by Theorem 3.

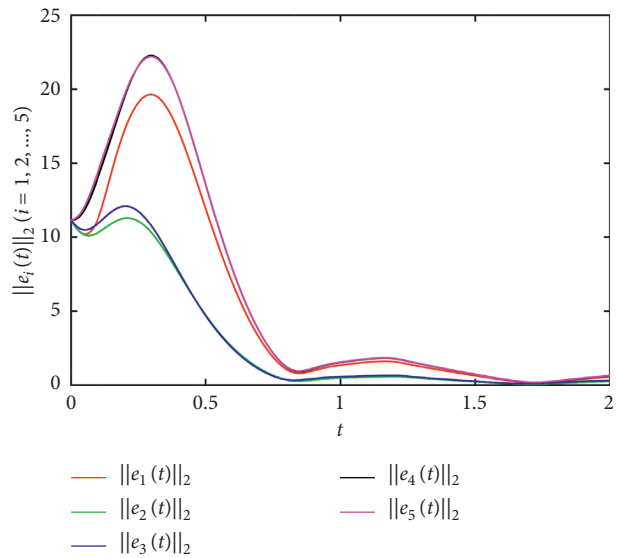


Figure 5: The changes of $\|e_i(t)\|_2 (i = 1, 2, \ldots, 5)$ for the system under the control strategy designed by Theorem 4.

Figure 5 also verifies that, under the control strategy designed by Theorem 4, the entire system gradually converges to a limited area.

5. Conclusions

The decentralized adaptive control for QC of HMASs has been studied. The combined adaptation of the coupling weights and control gains allows to drive HMASs (2) and (3) to some bounded areas. In addition, some pinning schemes have been proposed to adjust a fraction of the coupling weights and control gains. To deal with the heterogeneity, two new lemmas are proposed to derive the QC criteria. It has been shown that the QC can be obtained without requiring any global. In future works, we will extend the results to more general HMASs, such as fractional-order HMASs, HMASs with time delay, and cooperative-competitive interaction. At the same time, we will attempt to optimize control protocol and extend it to some other systems, i.e., fractional-order systems [48], memristive neural networks [49, 50], and complex-valued neural networks [51, 52].

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported jointly by the "Chunhui Plan" Cooperative Research for Ministry of Education, under Grant 191657, the Foundation of Science and Technology Department of Sichuan Province, under Grant 2020ZHCG0076, the Key Scientific Research Fund Project of Xihua University, under Grant Z17124, the Graduate Innovation Fund of Xihua University, under Grant YCJJ2019035, the Open Research Subject of Artificial Intelligence Key Laboratory of Sichuan Province, under Grant 2017RYJ03, the Key Research and Development Project of Sichuan Province, under Grant 2021YFG0071, and the Major Scientific and Technological Innovation Project of Chengdu, Sichuan Province, under Grant 2019-YF08-00003-GX.

References

[1] M. Xu, K. An, L. H. Vu, Z. Ye, J. Feng, and E. Chen, "Optimizing multi-agent based urban traffic signal control system," *Journal of Intelligent Transportation Systems*, vol. 23, no. 4, pp. 357–369, 2019.

[2] S.-X. Tang, J. Qi, and J. Zhang, "Formation tracking control for multi-agent systems: a wave-equation based approach," *International Journal of Control, Automation and Systems*, vol. 15, no. 6, pp. 2704–2713, 2017.

[3] S. M. Muyeen, A. Ghosh, S. M. Islam, and M. S. Baptista, "Multi-agent systems in ICT enabled smart grid: a status update on technology framework and applications," *IEEE Access*, vol. 7, pp. 97959–97973, 2019.

[4] P. P. Li, S. Yang, and S. C. Wang, "A decentralized multi-agent control approach for robust robot plan execution," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, pp. 1–14, 2018.

[5] J. Qin and Q. Ma, "Recent advances in consensus of multi-agent systems: a brief survey," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, 2016.

[6] Z.-H. Zhu, Z.-H. Guan, B. Hu, D.-X. Zhang, X.-M. Cheng, and T. Li, "Semi-global bipartite consensus tracking of singular multi-agent systems with input saturation," *Neurocomputing*, vol. 432, pp. 183–193, 2021.

[7] J. Ni, P. Shi, Y. Zhao, and Z. Wu, "Fixed-time output consensus tracking for high-order multi-agent systems with directed network topology and packet dropout," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 817–836, 2021.

[8] X. Guo, J. Liang, and J. Lu, "Scaled consensus problem for multi-agent systems with semi-Markov switching topologies: a view from the probability," *Journal of the Franklin Institute*, vol. 358, no. 6, pp. 3150–3166, 2021.

[9] C. Du, X. Liu, W. Ren, P. Lu, and H. Liu, "Finite-time consensus for linear multiagent systems via event-triggered strategy without continuous communication," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 19–29, 2020.

[10] Y. Su and J. Huang, "Two consensus problems for discrete-time multi-agent systems with switching network topology," *Automatica*, vol. 48, no. 9, pp. 1988–1997, 2012.

[11] T. Wang, M. Hu, and Y. Zhao, "Consensus of linear multi-agent systems with stochastic noises and binary-valued communications," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 13, pp. 4863–4879, 2020.

[12] Q. Shen, P Shi, J Zhu, S Wang, and Y Shi, "Neural networks-based distributed adaptive control of nonlinear multiagent systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 1010–1021, 2020.

[13] H. Su, Y. Liu, and Z. Zeng, "Second-order consensus for multiagent systems via intermittent sampled position data control," *IEEE Transactions on Cybernetics*, vol. 50, no. 5, pp. 2063–2072, 2020.

[14] M. S. Ali, R. Agalya, Z. Orman, and S. Arik, "Leader-following consensus of non-linear multi-agent systems with interval time-varying delay via impulsive control," *Neural Processing Letters*, vol. 53, no. 1, pp. 69–83, 2020.

[15] Z. Wei, J. Huang, and P. Wei, "Weak synchronization of chaotic neural networks with parameter mismatch via periodically intermittent control," *Applied Mathematical Modelling*, vol. 35, no. 2, pp. 612–620, 2011.

[16] H. Kim, H. Shim, and J. H. Seo, "Output consensus of heterogeneous uncertain linear multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 200–206, 2010.

[17] G. Wen, Y. Zhang, Z. Peng, Y. Yu, and A. Rahmani, "Observer-based output consensus of leader-following fractional-order heterogeneous nonlinear multi-agent systems," *International Journal of Control*, vol. 93, no. 10, pp. 2516–2524, 2020.

[18] S. Luo, X. Xu, L. Liu, and G. Feng, "Output consensus of heterogeneous linear multi-agent systems with communication, input and output time-delays," *Journal of the Franklin Institute*, vol. 357, no. 17, pp. 12825–12839, 2020.

[19] Y.-Y. Qian, L. Liu, and G. Feng, "Output consensus of heterogeneous linear multi-agent systems with adaptive event-triggered control," *IEEE Transactions on Automatic Control*, vol. 64, no. 6, pp. 2606–2613, 2019.

[20] J. Han, H. Zhang, H. Jiang et al., "H∞ consensus for linear heterogeneous multi-agent systems with state and output feedback control," *Neurocomputing*, vol. 18, no. 10, pp. 2468–2481, 2020.

[21] W. Hu, L. Liu, and G. Feng, "Output consensus of heterogeneous linear multi-agent systems by distributed event-triggered/self-triggered strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1914–1924, 2017.

[22] Q. Ma and G. Miao, "Output consensus for heterogeneous multi-agent systems with linear dynamics," *Applied Mathematics and Computation*, vol. 271, pp. 548–555, 2015.

[23] Y. Cai, H. Zhang, Z. Gao, and S. Sun, "The distributed output consensus control of linear heterogeneous multi-agent systems based on event-triggered transmission mechanism under directed topology," *Journal of the Franklin Institute*, vol. 357, no. 6, pp. 3267–3298, 2020.

[24] Z. Meng, Z. Lin, and W. Ren, "Robust cooperative tracking for multiple non-identical second-order nonlinear systems," *Automatica*, vol. 49, no. 8, pp. 2363–2372, 2013.

[25] P. Gong and W. Lan, "Adaptive robust tracking control for uncertain nonlinear fractional-order multi-agent systems with directed topologies," *Automatica*, vol. 92, pp. 92–99, 2018.

[26] J. Sun and Z. Geng, "Adaptive consensus tracking for linear multi-agent systems with heterogeneous unknown nonlinear dynamics," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 1, pp. 154–173, 2016.

[27] C.-J. Li and G.-P. Liu, "Consensus for heterogeneous networked multi-agent systems with switching topology and time-varying delays," *Journal of the Franklin Institute*, vol. 355, no. 10, pp. 4198–4217, 2018.

[28] C. D. Cruz-Ancona, R. Martinez-Guerra, and C. A. Perez-Pinacho, "A Leader-following consensus problem of multi-agent systems in heterogeneous networks," *Automatica*, vol. 115, pp. 108–899, 2020.

[29] Z. Wang and J. Cao, "Quasi-consensus of second-order leader-following multi-agent systems," *IET Control Theory & Applications*, vol. 6, no. 4, pp. 545–551, 2012.

[30] L. Wang, W. Qian, and Q.-G. Wang, "Bounded synchronisation of a time-varying dynamical network with nonidentical nodes," *International Journal of Systems Science*, vol. 46, no. 7, pp. 1234–1245, 2013.

[31] H. Yang, Z. Wang, M. Xiao, G.-P. Jiang, and C. Huang, "Quasi-synchronization of heterogeneous dynamical networks with sampled-data and input saturation," *Neurocomputing*, vol. 339, pp. 130–138, 2019.

[32] L. Pan and J. Cao, "Stochastic quasi-synchronization for delayed dynamical networks via intermittent control," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 3, pp. 1332–1343, 2012.

[33] F. Wang, Z. Zheng, and Y. Yang, "Quasi-synchronization of heterogenous fractional-order dynamical networks with time-varying delay via distributed impulsive control," *Chaos, Solitons and Fractals*, vol. 142, pp. 110–465, 2021.

[34] W. Fei and Y. Yang, "Quasi-synchronization for fractional-order delayed dynamical networks with heterogeneous nodes," *Applied Mathematics and Computation*, vol. 339, pp. 1–14, 2018.

[35] Z. Wang, J. Fan, G. P. Jiang et al., "Consensus in nonlinear multi-agent systems with nonidentical nodes and sampled-data control," *Science China-Information Sciences*, vol. 61, no. 12, pp. 122–203, 2018.

[36] W. Zhang, D. W. C. Ho, and Y. Liu, "Quasi-consensus of heterogeneous-switched nonlinear multiagent systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3136–3146, 2020.

[37] D. Ye and Y. Shao, "Quasi-synchronization of heterogeneous nonlinear multi-agent systems subject to DOS attacks with impulsive effects," *Neurocomputing*, vol. 366, pp. 131–139, 2019.

[38] W. Yu, G. Chen, and M. Cao, "Delay-induced consensus and quasi-consensus in multi-agent dynamical systems," *IEEE Transactions on Circuits & Systems I Regular Papers*, vol. 60, no. 10, pp. 2679–2687, 2010.

[39] Q. Xu, S. Zhuang, S. Liu, and J. Xiao, "Decentralized adaptive coupling synchronization of fractional-order complex-variable dynamical networks," *Neurocomputing*, vol. 186, pp. 119–126, 2016.

[40] Q. Xu, X. Xu, S. Zhuang, J. Xiao, C. Song, and C. Che, "New complex projective synchronization strategies for drive-response networks with fractional complex-variable dynamics," *Applied Mathematics and Computation*, vol. 338, pp. 552–566, 2018.

[41] J.-L. Wang, H.-N. Wu, T. Huang, S.-Y. Ren, and J. Wu, "Passivity analysis of coupled reaction-diffusion neural networks with Dirichlet boundary conditions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2148–2159, 2017.

[42] J.-L. Wang, H.-N. Wu, T. Huang, S.-Y. Ren, and J. Wu, "Passivity and output synchronization of complex dynamical networks with fixed and adaptive coupling strength," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 364–376, 2018.

[43] J.-L. Wang, H.-N. Wu, T. Huang, S.-Y. Ren, and J. Wu, "Passivity of directed and undirected complex dynamical networks with adaptive coupling weights," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 8, pp. 1827–1839, 2017.

[44] Q. Shen, P. Shi, J. Zhu et al., "Neural networks-based distributed adaptive control of nonlinear multiagent systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 1010–1021, 2019.

[45] Y. Fan, T. Xiao, and Z. Li, "Distributed fuzzy adaptive control for heterogeneous nonlinear multiagent systems with similar composite structure," *Complexity*, vol. 2020, Article ID 4081904, 10 pages, 2020.

[46] W. Yu, P. DeLellis, G. Chen, M. Di Bernardo, and J. Kurths, "Distributed adaptive control of synchronization in complex networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2153–2158, 2012.

[47] D. D'Angeli and A. Donno, "Shuffling matrices, kronecker product and discrete fourier transform," *Discrete Applied Mathematics*, vol. 233, pp. 1–18, 2017.

[48] Q. Xu, S. Zhuang, X. Xu, C. Che, and Y. Xia, "Stabilization of a class of fractional-order nonautonomous systems using quadratic Lyapunov functions," *Advances in Difference Equations*, vol. 2018, pp. 1–15, 2018.

[49] Y. Huang, S. Lin, and E. Yang, "Event-triggered passivity of multi-weighted coupled delayed reaction-diffusion memristive neural networks with fixed and switching topologies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 89, pp. 1–28, 2020.

[50] Y.-L. Huang, S.-H. Qiu, and S.-Y. Ren, "Finite-time synchronisation and passivity of coupled memristive neural networks," *International Journal of Control*, vol. 93, no. 12, pp. 2824–2837, 2020.

[51] X. Xu, Q. Xu, J. Yang, H. Xue, and Y. Xu, "Further research on exponential stability for quaternion-valued neural networks with mixed delays," *Neurocomputing*, vol. 400, pp. 186–205, 2020.

[52] X. Xu, J. Yang, Q. Xu, Y. Xu, and S. Sun, "Exponential stability for delayed complex-valued neural networks with reaction-diffusion terms," *Advances in Difference Equations*, vol. 2021, no. 1, pp. 1–27, 2021.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 9905125, 12 pages
https://doi.org/10.1155/2021/9905125

Research Article

Finite-Time Passivity of Stochastic Coupled Complex Networks

Xunwu Yin [1] and Min Cao [2]

[1] *School of Mathematical Sciences, Tiangong University, Tianjin 300387, China*
[2] *School of Computer Science and Technology, Tiangong University, Tianjin 300387, China*

Correspondence should be addressed to Xunwu Yin; yinxunwu@163.com

Received 30 April 2021; Accepted 13 June 2021; Published 9 July 2021

Academic Editor: A. E. Matouk

The finite-time passivity problem is, respectively, investigated for stochastic coupled complex networks (SCCNs) with and without time-varying delay. Firstly, we present several new concepts about finite-time passivity in the sense of expectation on the basis of existing passivity definition. By designing appropriate controllers, the finite-time passivity of SCCNs with and without time-varying delay is obtained. In addition, the definition of finite-time synchronization in the sense of expectation is proposed. Under some sufficient conditions and designed controllers, finite-time passivity derives finite-time synchronization. Finally, two examples are given to demonstrate the effectiveness of finite-time passive and synchronization criteria.

1. Introduction

In the real world, complex networks can be seen everywhere such as food webs, communication networks, World Wide Web, and many others [1–3]. Due to various uncertainties in the actual system, complex network systems may be affected by noise. In recent years, the stability of stochastic systems has been extensively studied. At the same time, the synchronization and stability of stochastic complex networks have gradually become a topic of widespread concern for scholars in various fields [4–8].

Passivity is one part of dissipativeness. The main property of passivity is keeping the systems internally stable. The passivity theory has been extensively applied in many fields such as stability, complexity, signal processing, chaos control, synchronization fuzzy control, and so on [9–12]. These are the main reasons why the passivity theory has been one of the most active research areas. In [13], the problem of passivity analysis was studied for discrete-time stochastic Markovian jump neural networks with both discrete and distributed delays. In [14], the problem of passivity analysis is investigated for a class of discrete-time stochastic neural networks with time-varying delays.

It is well known that passivity theory can provide a powerful tool to analyze synchronization of complex networks. However, in many existing works, synchronization is defined over the infinite time interval. Most of the theoretical methods on the synchronization of complex networks can only realize the network [15] or exponential asymptotical synchronization [16] which guarantees that error tends to 0 when t tends to infinity. That is to say, achieving asymptotically stable convergence will be in infinite time. No further consideration has been given to the time and speed of synchronization. However, in practical engineering, people usually expect faster convergence rate and predict the required convergence time. Consequently, in order to achieve better control, the idea of finite-time synchronization has been proposed, and more and more attention has been paid by researchers. This kind of method can predict the synchronization time in advance and has better robustness, anti-interference, and better control effect. It has important research significance in theory and practice. Therefore, it is more meaningful to study finite-time synchronization [17–21]. In [22], the authors study finite-time passivity of multi-weighted coupled neural networks with and without coupling delays. As far as we know, very few scholars have discussed finite-time passivity of stochastic complex networks in recent years.

Motivated by the above discussions, we will investigate finite-time passivity of stochastic coupled complex networks (SCCNs). The main novelty and contributions of this paper can be summarized as follows. Firstly, we give three concepts of finite-time passivity in the sense of expectation. Secondly, we develop several finite-time passivity criteria. Lastly, we establish the relationship between finite-time passivity and finite-time synchronization in the sense of expectation.

2. Lemmas and Definitions

In this section, we will give some lemmas and definitions.

2.1. Lemmas

Lemma 1 (see [23]). *Assume that a continuous, positive-definite function $W(t)$ satisfies the following differential inequality:*

$$\dot{W}(t) \leq -\varrho W^{\mu}(t), \quad t \geq t_0, W(t_0) \geq 0, \tag{1}$$

where $\varrho > 0$ and $0 < \mu < 1$ are constants. Then, for any given t_0, $W(t)$ satisfies the following inequality:

$$\begin{aligned} &W^{1-\mu}(t) \leq W^{1-\mu}(t_0) - \varrho(1-\mu)(t-t_0), \quad t_0 \leq t \leq t_1, \\ &W(t) \equiv 0, \quad t \geq t_1, \end{aligned} \tag{2}$$

with t_1 given by

$$t_1 = t_0 + \frac{W^{1-\mu}(t_0)}{\varrho(1-\mu)}. \tag{3}$$

Lemma 2 (see [24]). *For any $b_i \in \mathbb{R}, i = 1, \ldots, n, 0 < p \leq 1$, the following inequality holds:*

$$\left(\sum_{i=1}^{n} |b_i|\right)^p \leq \sum_{i=1}^{n} |b_i|^p. \tag{4}$$

Lemma 3 (see [25]). *For any vectors $x, y \in \mathbb{R}^n$ and matrix $0 < P \in \mathbb{R}^{n\times n}$, the following inequality holds:*

$$x^T y + y^T x \leq x^T P x + y^T P^{-1} y. \tag{5}$$

2.2. Definitions. Next, we will give three definitions about finite-time passivity in the sense of expectation. $E\{\cdot\}$ in these definitions stands for the mathematical expectation operator with respect to the given probability.

Definition 1. A stochastic system with input $u(t) \in \mathbb{R}^n$ and output $y(t) \in \mathbb{R}^n$ is said to be finite-time passive in the sense of expectation if there exists a nonnegative function V such that

$$E\left\{u^T(t)y(t)\right\} \geq \frac{E\{\mathrm{d}V(t)\}}{\mathrm{d}t} + \beta E\{V^{\alpha}(t)\}, \tag{6}$$

for some $\alpha \in (0,1)$ and $\beta > 0$.

Definition 2. A stochastic system with input $u(t) \in \mathbb{R}^n$ and output $y(t) \in \mathbb{R}^n$ is finite-time input strictly passive in the sense of expectation if there exists a nonnegative function V such that

$$E\left\{u^T(t)y(t)\right\} - \gamma_1 E\left\{u^T(t)u(t)\right\} \geq \frac{E\{\mathrm{d}V(t)\}}{\mathrm{d}t} + \beta E\{V^{\alpha}(t)\}, \tag{7}$$

for some $\alpha \in (0,1), \beta > 0$, and $\gamma_1 > 0$.

Definition 3. A stochastic system with input $u(t) \in \mathbb{R}^n$ and output $y(t) \in \mathbb{R}^n$ is finite-time output strictly passive in the sense of expectation if there exists a nonnegative function V such that

$$E\left\{u^T(t)y(t)\right\} - \gamma_2 E\left\{y^T(t)y(t)\right\} \geq \frac{E\{\mathrm{d}V(t)\}}{\mathrm{d}t} + \beta E\{V^{\alpha}(t)\}, \tag{8}$$

for some $\alpha \in (0,1), \beta > 0$, and $\gamma_2 > 0$.

Definition 4 (see [25]). Let $A = (a_{ij})_{m\times m} \in \mathbb{R}^{m\times n}$ and $B = (b_{ij})_{p\times q} \in \mathbb{R}^{p\times q}$. Then, the Kronecker product of A and B is defined as the matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{2n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} \in \mathbb{R}^{mp\times nq}. \tag{9}$$

Throughout this paper, we make the following assumptions.

(H1) (see [26]) The function $f(\cdot)$ is in the QUAD class, that is, there exist diagonal matrices $0 < P = \mathrm{diag}(p_1, p_2, \ldots, p_n) \in \mathbb{R}^{n\times n}$ and $\Delta = \mathrm{diag}(\delta_1, \delta_2, \ldots, \delta_n) \in \mathbb{R}^{n\times n}$, such that

$$(x-y)^T P[f(x) - f(y) - \Delta(x-y)] \leq -\lambda(x-y)^T(x-y), \tag{10}$$

for all $x, y \in \mathbb{R}^n$ and some $\lambda > 0$.

(H2) For arbitrary $u, v \in \mathbb{R}^n$, there exists a positive constant L such that the following inequality holds:

$$\mathrm{trace}[h(u) - h(v)]^T[h(u) - h(v)] \leq L(u-v)^T(u-v). \tag{11}$$

Remark 1 (see [25]). It can be verified that many of the benchmark chaotic systems belong to "function class QUAD," such as the Lorenz system, the Chen system, and the *Lü* system.

3. Finite-Time Passivity of SCCNs

3.1. Network Model. In this paper, we will consider the following stochastic coupled complex networks model:

$$\mathrm{d}z_i(t) = \left[f(z_i(t)) + a\sum_{j=1}^{N} G_{ij}\Gamma z_j(t) + u_i(t) + v_i(t) \right]\mathrm{d}t + h(z_i(t))\mathrm{d}\omega(t), \tag{12}$$

where $z_i(t) = (z_{i1}(t), z_{i2}(t), \ldots, z_{in}(t))^T \in \mathbb{R}^n$ is the state vector of the ith node; N corresponds to the number of neurons; $f(z_i(t)) = (f_1(z_{i1}(t)), f_2(z_{i2}(t)), \ldots, f_n(z_{in}(t)))^T \in \mathbb{R}^n$ denotes the neuron activation function and satisfies assumption (H1); $u_i(t) = (u_{i1}(t), u_{i2}(t), \ldots, u_{in}(t))^T \in \mathbb{R}^n$ is a varying external input vector; $v_i(t) = (v_{i1}(t), v_{i2}(t), \ldots, v_{in}(t))^T \in \mathbb{R}^n$ denotes the control input; $h(\cdot) \in \mathbb{R}^{n\times n}$ satisfies assumption (H2); $\omega(t) = (\omega_1(t), \omega_2(t), \ldots, \omega_n(t))^T \in \mathbb{R}^n$ is a n–dimensional Brownian motion defined on a complete probability space (Ω, P); a is a positive real number which represents the overall coupling strength; Γ denotes the inner coupling matrix; and $G = (G_{ij})_{N\times N}$ represents the topological structure of the network, where G_{ij} is defined as follows: if there exists a connection between node i and node j, then $G_{ij} = G_{ji} > 0$; otherwise, $G_{ij} = G_{ji} = 0$, $(i \neq j)$, and the diagonal elements of matrix G are defined by

$$G_{ii} = -\sum_{\substack{j=1\\ j\neq i}}^{N} G_{ij}, \quad i = 1, 2, \ldots, N. \tag{13}$$

3.2. Finite-Time Passivity. Set synchronization function $\overline{z}(t)$ satisfies

$$\mathrm{d}\overline{z}(t) = f(\overline{z}(t))\mathrm{d}t + h(\overline{z}(t))\mathrm{d}\omega(t), \tag{14}$$

where $\overline{z}(t) = (\overline{z}_1(t), \overline{z}_2(t), \ldots, \overline{z}_n(t)) \in \mathbb{R}^n$.

Define $e_i(t) = z_i(t) - \overline{z}(t)$, $i = 1, 2, \ldots, N$. Then, we have

$$\begin{aligned}\mathrm{d}e_i(t) = {} & \Big[f(z_i(t)) - f(\overline{z}(t)) \\ & + a\sum_{j=1}^{N} G_{ij}\Gamma e_j(t) + u_i(t) + v_i(t) \Big]\mathrm{d}t \\ & + \left[h(z_i(t)) - h(\overline{z}(t)) \right]\mathrm{d}\omega(t),\end{aligned} \tag{15}$$

where $i = 1, 2, \ldots, N$.

$y_i(t) \in \mathbb{R}^n$ refers to the output vector of (15) and is defined as follows:

$$y_i(t) = A_1 e_i(t) + A_2 u_i(t), \tag{16}$$

where $A_1, A_2 \in \mathbb{R}^{n\times n}$,

The controller for network (12) is defined as follows:

$$\begin{aligned} v_i(t) = {} & -Q_i(z_i(t) - \overline{z}(t)) \\ & - \beta P^{((\alpha-1)/2)} \operatorname{sign}(z_i(t) - \overline{z}(t)) \left| z_i(t) - \overline{z}(t) \right|^{\alpha}, \end{aligned} \tag{17}$$

where $Q_i \in \mathbb{R}^{n\times n}$, $0 < \alpha < 1, \beta > 0, P$ is defined in (H1), and

$$\begin{aligned} \operatorname{sign}(z_i(t) - \overline{z}(t)) &= \operatorname{diag}\left(\operatorname{sign}(z_{i1}(t) - \overline{z}_1(t)), \operatorname{sign}(z_{i2}(t) - \overline{z}_2(t)), \ldots, \operatorname{sign}(z_{in}(t) - \overline{z}_n(t))\right), \\ \left| z_i(t) - \overline{z}(t) \right|^{\alpha} &= \left(\left| z_{i1}(t) - \overline{z}_1(t) \right|^{\alpha}, \left| z_{i2}(t) - \overline{z}_2(t) \right|^{\alpha}, \ldots, \left| z_{in}(t) - \overline{z}_n(t) \right|^{\alpha} \right)^T, \\ P^{((\alpha-1)/2)} &= \operatorname{diag}\left(p_1^{((\alpha-1)/2)}, p_2^{((\alpha-1)/2)}, \ldots, p_n^{((\alpha-1)/2)} \right). \end{aligned} \tag{18}$$

Theorem 1. *Under assumptions (H1) and (H2), network model (15) is finite-time passive in the sense of expectation under controller (17) if there exists matrix* $Q = diag(Q_1, Q_2, \ldots, Q_N) \in \mathbb{R}^{nN\times nN}$ *such that*

$$\begin{pmatrix} K_1 & E_1 \\ E_1^T & -I_N \otimes \dfrac{A_2 + A_2^T}{2} \end{pmatrix} \leq 0, \tag{19}$$

where

$$\begin{aligned} K_1 &= I_N \otimes (P\Delta + \Delta P - 2\lambda I_n + \lambda_0 I_n) - (I_N \otimes P)Q \\ &\quad - Q^T(I_N \otimes P) + aG \otimes (P\Gamma + \Gamma P), \\ E_1 &= I_N \otimes P - \frac{I_N \otimes A_1^T}{2}. \end{aligned} \tag{20}$$

Proof. For network (15), the Lyapunov functional is chosen as follows:

$$V_1(t) = \sum_{i=1}^{N} e_i^T(t) P e_i(t). \tag{21}$$

According to Ito's lemma, we acquire from (15) and (17)

$$\mathrm{d}V_1(t) = \mathscr{L}V_1(t)\mathrm{d}t + \Phi(t)\mathrm{d}\omega(t). \tag{22}$$

Here

$$\begin{aligned} \mathscr{L}V_1(t) = {} & 2\sum_{i=1}^{N} e_i^T(t) P \Big[f(z_i(t)) - f(\overline{z}(t)) + a\sum_{j=1}^{N} G_{ij}\Gamma e_j(t) \\ & + u_i(t) - Q_i e_i(t) - \beta P^{((\alpha-1)/2)} \operatorname{sign}(e_i(t)) \left| e_i(t) \right|^{\alpha} \Big] \\ & + \sum_{i=1}^{N} \operatorname{trace}\left[h(z_i(t)) - h(\overline{z}(t)) \right]^T P \left[h(z_i(t)) - h(\overline{z}(t)) \right], \\ \Phi(t) = {} & 2\sum_{i=1}^{N} e_i^T(t) P \left[h(z_i(t)) - h(\overline{z}(t)) \right]. \end{aligned} \tag{23}$$

According to (H1), we can obtain

$$\sum_{i=1}^{N} e_i^T(t)P\left(f\left(z_i(t)\right)-f\left(\overline{z}(t)\right)\right)\leq \sum_{i=1}^{N} e_i^T(t)\left(P\Delta-\lambda I_n\right)e_i(t). \tag{24}$$

We can get the following from (H2):

$$\begin{aligned}&\operatorname{trace}\left[h\left(z_i(t)\right)-h\left(\overline{z}(t)\right)\right]^T P\left[h\left(z_i(t)\right)-h\left(\overline{z}(t)\right)\right]\\&\quad\leq \lambda_M(P)\operatorname{trace}\left[h\left(z_i(t)\right)-h\left(\overline{z}(t)\right)\right]^T\left[h\left(z_i(t)\right)-h\left(\overline{z}(t)\right)\right]\\&\quad\leq L\lambda_M(P)\left[z_i(t)-\overline{z}(t)\right]^T\left[z_i(t)-\overline{z}(t)\right]=\lambda_0 e_i^T(t)e_i(t).\end{aligned} \tag{25}$$

Here $\lambda_M(P)$ represents maximum eigenvalue of matrix P, $\lambda_0 = L\lambda_M(P)$.

Thus,

$$\begin{aligned}\mathscr{L}V_1(t)\leq&\sum_{i=1}^{N} e_i^T(t)\left(P\Delta+\Delta P-2\lambda I_n+\lambda_0 I_n\right)e_i(t)\\&+2a\sum_{i=1}^{N}\sum_{j=1}^{N} G_{ij}e_i^T(t)P\Gamma e_j(t)\\&+2\sum_{i=1}^{N} e_i^T(t)Pu_i(t)-2\sum_{i=1}^{N} e_i^T(t)PQ_ie_i(t)\\&-2\beta\sum_{i=1}^{N} e_i^T(t)P^{((\alpha+1)/2)}\operatorname{sign}\left(e_i(t)\right)\left|e_i(t)\right|^{\alpha}\\=&\,e^T(t)\left[I_N\otimes\left(P\Delta+\Delta P-2\lambda I_n+\lambda_0 I_n\right)\right.\\&-\left(I_N\otimes P\right)Q-Q^T\left(I_N\otimes P\right)\\&\left.+aG\otimes(P\Gamma+\Gamma P)\right]e(t)+2e^T(t)\left(I_N\otimes P\right)u(t)\\&-2\beta\sum_{i=1}^{N} e_i^T(t)P^{((\alpha+1)/2)}\operatorname{sign}\left(e_i(t)\right)\left|e_i(t)\right|^{\alpha},\end{aligned} \tag{26}$$

where $e(t)=(e_1^T(t),e_2^T(t),\ldots,e_N^T(t))^T$, $u(t)=(u_1^T(t),u_2^T(t),\ldots,u_N^T(t))^T$.

Considering $\operatorname{sign}x\cdot x=|x|$, $(\forall x\in\mathbb{R})$ and Lemma 2, we can easily conclude that

$$\begin{aligned}&\sum_{i=1}^{N} e_i^T(t)P^{((\alpha+1)/2)}\operatorname{sign}\left(e_i(t)\right)\left|e_i(t)\right|^{\alpha}\\&\quad=\sum_{i=1}^{N}\sum_{j=1}^{n} p_j^{((\alpha+1)/2)}\left|e_{ij}(t)\right|^{\alpha+1}\\&\quad\geq\sum_{i=1}^{N}\left(\sum_{j=1}^{n} p_je_{ij}^2(t)\right)^{((\alpha+1)/2)}\\&\quad=\sum_{i=1}^{N}\left(e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)}.\end{aligned} \tag{27}$$

Set $y(t)=(y_1^T(t),y_2^T(t),\ldots,y_N^T(t))^T$; consequently,

$$\begin{aligned}u^T(t)y(t)&=\sum_{i=1}^{N} u_i^T(t)y_i(t)\\&=\sum_{i=1}^{N} u_i^T(t)\left[A_1e_i(t)+A_2u_i(t)\right]\\&=u^T(t)\left(I_N\otimes A_1\right)e(t)\\&\quad+u^T(t)\left(I_N\otimes A_2\right)u(t).\end{aligned} \tag{28}$$

From (19) and (26)–(28),

$$\begin{aligned}&\mathscr{L}V_1(t)-u^T(t)y(t)\\&\quad\leq-2\beta\sum_{i=1}^{N}\left(e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)}\\&\qquad+\zeta^T(t)\begin{pmatrix}K_1 & E_1\\ E_1^T & -I_N\otimes\dfrac{A_2+A_2^T}{2}\end{pmatrix}\zeta(t)\\&\quad\leq-2\beta\sum_{i=1}^{N}\left(e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)}\\&\quad\leq-2\beta\left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)}\\&\quad=-2\beta V_1^{((\alpha+1)/2)}(t),\end{aligned} \tag{29}$$

where $\zeta(t)=(e^T(t),u^T(t))^T$.

Considering $E\{\mathrm{d}V_1(t)\}=E\{\mathscr{L}V_1(t)\mathrm{d}t\}$, consequently

$$\begin{aligned}&E\left\{\mathrm{d}V_1(t)-u^T(t)y(t)\mathrm{d}t\right\}\\&\quad=E\left\{\left[\mathscr{L}V_1(t)-u^T(t)y(t)\right]\mathrm{d}t\right\}\\&\quad\leq-2\beta E\left\{V_1^{((\alpha+1)/2)}(t)\mathrm{d}t\right\}.\end{aligned} \tag{30}$$

Then, we can obtain

$$E\left\{u^T(t)y(t)\right\}\geq\frac{E\{\mathrm{d}V_1(t)\}}{\mathrm{d}t}+2\beta E\left\{V_1^{((\alpha+1)/2)}(t)\right\}. \tag{31}$$

Therefore, network (15) is finite-time passive in the sense of expectation under controller (17). □

Theorem 2. *Under assumptions (H1) and (H2), network model (15) is finite-time input strictly passive in the sense of expectation under controller (17) if there exist matrix $Q=\operatorname{diag}(Q_1,Q_2,\ldots,Q_N)\in\mathbb{R}^{nN\times nN}$ and a positive real number γ_1 such that*

$$\begin{pmatrix}K_1 & E_1\\ E_1^T & \gamma_1 I_{nN}-I_N\otimes\dfrac{A_2+A_2^T}{2}\end{pmatrix}\leq 0, \tag{32}$$

where K_1, E_1 have the same meanings as in Theorem 1.

Proof. We will choose the same $V_1(t)$ as (21) for network (15).

By (26)–(28), one can get

$$\begin{aligned}
&\mathscr{L}V_1(t) - u^T(t)y(t) + \gamma_1 u^T(t)u(t) \\
&\le e^T(t)\big[I_N \otimes (P\Delta + \Delta P - 2\lambda I_n + \lambda_0 I_n) \\
&\quad - (I_N \otimes P)Q - Q^T(I_N \otimes P) \\
&\quad + aG \otimes (P\Gamma + \Gamma P)\big]e(t) \\
&\quad + 2e^T(t)(I_N \otimes P)u(t) + \gamma_1 u^T(t)u(t) \\
&\quad - 2\beta \sum_{i=1}^{N} e_i^T(t)P^{((\alpha+1)/2)}\operatorname{sign}(e_i(t))|e_i(t)|^{\alpha} \\
&\quad - u^T(t)(I_N \otimes A_1)e(t) - u^T(t)(I_N \otimes A_2)u(t) \\
&= -2\beta \sum_{i=1}^{N} e_i^T(t)P^{((\alpha+1)/2)}\operatorname{sign}(e_i(t))|e_i(t)|^{\alpha} \\
&\quad + \zeta^T(t)\begin{pmatrix} K_1 & E_1 \\ E_1^T & \gamma_1 I_{nN} - I_N \otimes \dfrac{A_2 + A_2^T}{2} \end{pmatrix}\zeta(t) \\
&\le -2\beta\left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)} = -2\beta V_1^{((\alpha+1)/2)}(t).
\end{aligned} \tag{33}$$

Taking the mathematical expectation on both sides above, one can derive that

$$\begin{aligned}
&E\{u^T(t)y(t)\} - \gamma_1 E\{u^T(t)u(t)\} \\
&\ge \frac{E\{\mathrm{d}V_1(t)\}}{\mathrm{d}t} + 2\beta E\{V_1^{((\alpha+1)/2)}(t)\}.
\end{aligned} \tag{34}$$

Therefore, network (15) is finite-time input strictly passive in the sense of expectation under controller (17). □

Theorem 3. *Under assumptions (H1) and (H2), network model (15) is finite-time output strictly passive in the sense of expectation under controller (17) if there exist matrix $Q = diag(Q_1, Q_2, \ldots, Q_N) \in \mathbb{R}^{nN\times nN}$ and a positive real number γ_2 such that*

$$\begin{pmatrix} K_2 & E_2 \\ E_2^T & K_3 \end{pmatrix} \le 0, \tag{35}$$

where

$$\begin{aligned}
K_2 &= K_1 + \gamma_2\left(I_N \otimes A_1^T\right)(I_N \otimes A_1), \\
E_2 &= E_1 + \gamma_2\left(I_N \otimes A_1^T\right)(I_N \otimes A_2), \\
K_3 &= \gamma_2\left(I_N \otimes A_2^T\right)(I_N \otimes A_2) - I_N \otimes \frac{A_2 + A_2^T}{2},
\end{aligned} \tag{36}$$

K_1, E_1 have the same meanings as in Theorem 1.

Proof. Firstly we calculate the following equality:

$$\begin{aligned}
y^T(t)y(t) &= \sum_{i=1}^{N} y_i^T(t)y_i^T(t) \\
&= \sum_{i=1}^{N} [A_1 e_i(t) + A_2 u_i(t)]^T [A_1 e_i(t) + A_2 u_i(t)] \\
&= \sum_{i=1}^{N} \left[e_i^T(t)A_1^T + u_i^T(t)A_2^T\right][A_1 e_i(t) + A_2 u_i(t)] \\
&= \sum_{i=1}^{N} \left[e_i^T(t)A_1^T A_1 e_i(t) + e_i^T(t)A_1^T A_2 u_i(t)\right. \\
&\quad \left. + u_i^T(t)A_2^T A_1 e_i(t) + u_i^T(t)A_2^T A_2 u_i(t)\right] \\
&= e^T(t)\left(I_N \otimes A_1^T A_1\right)e(t) + e^T(t)\left(I_N \otimes A_1^T A_2\right)u(t) \\
&\quad + u^T(t)\left(I_N \otimes A_2^T A_1\right)e(t) + u^T(t)\left(I_N \otimes A_2^T A_2\right)u(t) \\
&= e^T(t)\left(I_N \otimes A_1^T\right)(I_N \otimes A_1)e(t) \\
&\quad + e^T(t)\left(I_N \otimes A_1^T\right)(I_N \otimes A_2)u(t) \\
&\quad + u^T(t)\left(I_N \otimes A_2^T\right)(I_N \otimes A_1)e(t) \\
&\quad + u^T(t)\left(I_N \otimes A_2^T\right)(I_N \otimes A_2)u(t).
\end{aligned} \tag{37}$$

For the last step, we utilize the important properties of the Kronecker product:

$$(M_1 \otimes M_2)(M_3 \otimes M_4) = (M_1 M_3) \otimes (M_2 M_4). \tag{38}$$

Select the same $V_1(t)$ as (21) for network (15). We can obtain

$$\begin{aligned}
&\mathscr{L}V_1(t) - u^T(t)y(t) + \gamma_2 y^T(t)y(t) \\
&\le e^T(t)\big[I_N \otimes (P\Delta + \Delta P - 2\lambda I_n + \lambda_0 I_n) \\
&\quad - (I_N \otimes P)Q - Q^T(I_N \otimes P) \\
&\quad + aG \otimes (P\Gamma + \Gamma P)\big]e(t) + 2e^T(t)(I_N \otimes P)u(t) \\
&\quad - u^T(t)(I_N \otimes A_1)e(t) - u^T(t)(I_N \otimes A_2)u(t) \\
&\quad + \gamma_2 e^T(t)\left(I_N \otimes A_1^T\right)(I_N \otimes A_1)e(t) \\
&\quad + \gamma_2 e^T(t)\left(I_N \otimes A_1^T\right)(I_N \otimes A_2)u(t) \\
&\quad + \gamma_2 u^T(t)\left(I_N \otimes A_2^T\right)(I_N \otimes A_1)e(t) \\
&\quad + \gamma_2 u^T(t)\left(I_N \otimes A_2^T\right)(I_N \otimes A_2)u(t) \\
&\quad - 2\beta \sum_{i=1}^{N} e_i^T(t)P^{((\alpha+1)/2)}\operatorname{sign}(e_i(t))|e_i(t)|^{\alpha} \\
&= -2\beta \sum_{i=1}^{N} e_i^T(t)P^{((\alpha+1)/2)}\operatorname{sign}(e_i(t))|e_i(t)|^{\alpha} \\
&\quad + \zeta^T(t)\begin{pmatrix} K_2 & E_2 \\ E_2^T & K_3 \end{pmatrix}\zeta(t) \\
&\le -2\beta\left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)} = -2\beta V_1^{((\alpha+1)/2)}(t).
\end{aligned} \tag{39}$$

Taking the mathematical expectation on both sides above, one can derive that

$$E\left\{u^T(t)y(t)\right\} - \gamma_2 E\left\{y^T(t)y(t)\right\} \geq \frac{E\{\mathrm{d}V_1(t)\}}{\mathrm{d}t} + 2\beta E\left\{V_1^{((\alpha+1)/2)}(t)\right\}. \tag{40}$$

Therefore, network (15) is finite-time output strictly passive in the sense of expectation under controller (17). □

3.3. Finite-Time Synchronization. In this section, we will verify finite-time synchronization in the sense of expectation for SCCNs (12). Firstly, the definition of finite-time synchronization is given as follows.

Definition 5. The SCCN (12) is finite-time synchronized in the sense of expectation if there exists a constant $T > 0$ such that

$$\lim_{t \longrightarrow T^-} E\left\{\left\|z_i(t) - \overline{z}(t)\right\|_2\right\} = 0, \quad E\left\{\left\|z_i(t) - \overline{z}(t)\right\|_2\right\} \equiv 0, \quad t \geq T, \tag{41}$$

for $i = 1, 2, \ldots, N$, where $u_i(t) = 0, i = 1, 2, \ldots, N$.

Theorem 4. *Assume that a continuous, positive-definite function $V(t)$ satisfies the following inequality:*

$$\varphi_1\left(E\|e(t)\|_2\right) \leq V(t), \tag{42}$$

where φ_1: $[0, +\infty) \longrightarrow [0, +\infty)$ is continuous and strictly monotonically increasing function and $\varphi_1(s)$ is positive for $s > 0$ with $\varphi_1(0) = 0$. If network (15) is finite-time passive (finite-time input strictly passive, finite-time output strictly passive) in the sense of expectation with respect to $V(t)$, then SCCN (12) is finite-time synchronized in the sense of expectation under controller (17).

Proof. The network model (15) is finite-time passive in the sense of expectation with respect to $V(t)$ under controller (17), that is to say, there exist $\alpha \in (0, 1)$ and $\beta > 0$ such that

$$E\left\{u^T(t)y(t)\right\} \geq \frac{E\{\mathrm{d}V(t)\}}{\mathrm{d}t} + \beta E\{V^\alpha(t)\}. \tag{43}$$

Considering $u(t) = 0$, one obtains

$$\frac{E\{\mathrm{d}V(t)\}}{\mathrm{d}t} + \beta E\{V^\alpha(t)\} \leq 0. \tag{44}$$

According to the property of mathematical expectation,

$$\dot{V}(t) \leq -\beta V^\alpha(t). \tag{45}$$

Choosing $t_0 = 0$ in Lemma 1, we can obtain $V(t) \equiv 0$ for $t \geqslant t_1$, where $t_1 = (V^{1-\alpha}(0)/\beta(1-\alpha))$. On the one hand, since

$$\varphi_1\left[E\left(\|e(t)\|_2\right)\right] \leq V(t), \tag{46}$$

one has

$$\varphi_1\left[E\left(\|e(t)\|_2\right)\right] \leq V(t) \equiv 0, \tag{47}$$

for $t \geq t_1$. Since $\varphi_1(s) = 0$ if and only if $s = 0$. Then, we can conclude that

$$E\left(\|e(t)\|_2\right) \equiv 0, \quad t \geq t_1. \tag{48}$$

On the other hand, $V(t)$ is continuous, so

$$\lim_{t \longrightarrow t_1^-} V(t) = \lim_{t \longrightarrow t_1^+} V(t) = 0. \tag{49}$$

Taking the limit $t \longrightarrow t_1^-$ on both sides of (46), we will get

$$\lim_{t \longrightarrow t_1^-} \varphi_1\left[E\left(\|e(t)\|_2\right)\right] = 0. \tag{50}$$

Namely, SCCN (12) is finite-time synchronized in the sense of expectation under controller (17).

Similarly, it is easy to prove that SCCN (12) is also finite-time synchronized in the sense of expectation under controller (17) if network model (15) is finite-time input strictly passive or finite-time output strictly passive in the sense of expectation. □

4. Finite-Time Passivity of SCCNs with Time-Varying Delay

4.1. Network Model. In this section, the network model is described by

$$\mathrm{d}z_i(t) = \left[f\left(z_i(t)\right) + a\sum_{j=1}^{N} G_{ij}\Gamma z_j(t - \tau(t)) + u_i(t) + v_i(t)\right]\mathrm{d}t + h\left(z_i(t)\right)\mathrm{d}\omega(t), \tag{51}$$

where $\tau(t)$ is the time delay and satisfies $0 \leq \tau(t) \leq \tau, 0 \leq \dot{\tau}(t) \leq d < 1$.

4.2. Finite-Time Passivity. Let $\overline{z}(t)$ also satisfy

$$\mathrm{d}\overline{z}(t) = f(\overline{z}(t))\mathrm{d}t + h(\overline{z}(t))\mathrm{d}\omega(t). \tag{52}$$

Define $e_i(t) = z_i(t) - \overline{z}(t)$, $i = 1, 2, \ldots, N$. Then, we have

$$\mathrm{d}e_i(t) = \left[f\left(z_i(t)\right) - f(\overline{z}(t)) + a\sum_{j=1}^{N} G_{ij}\Gamma e_j(t - \tau(t)) + u_i(t) + v_i(t)\right]\mathrm{d}t + \left[h\left(z_i(t)\right) - h(\overline{z}(t))\right]\mathrm{d}\omega(t). \tag{53}$$

The output vector $y_i(t)$ of network (53) is defined as follows:

$$y_i(t) = B_1 e_i(t) + B_2 u_i(t). \tag{54}$$

Design the following controller for network (51):

$$v_i(t) = -\beta P^{-1}\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t}\left(z_i(h)-\bar{z}(h)\right)^T M_i\left(z_i(h)-\bar{z}(h)\right)\mathrm{d}h\right)^{((\alpha+1)/2)}\frac{z_i(t)-\bar{z}(t)}{\left\|z_i(t)-\bar{z}(t)\right\|_2^2} - \beta P^{((\alpha-1)/2)}\operatorname{sign}\left(z_i(t)-\bar{z}(t)\right)\left|z_i(t)-\bar{z}(t)\right|^{\alpha} - Q_i\left(z_i(t)-\bar{z}(t)\right), \tag{55}$$

where $0 < M_i \in \mathbb{R}^{n\times n}$, $M = \operatorname{diag}(M_1, M_2, \ldots, M_N)$, Q_i, α, P, β, $P^{((\alpha-1)/2)}$, $\operatorname{sign}(z_i(t)-\bar{z}(t))$, $|z_i(t)-\bar{z}(t)|^{\alpha}$ have the same meanings as in (17).

Theorem 5. *Under assumptions (H1) and (H2), network model (53) is finite-time passive in the sense of expectation under controller (55) if there exist matrices* $Q = diag(Q_1, Q_2, \ldots, Q_N) \in \mathbb{R}^{nN\times nN}$, $0 < M = diag(M_1, M_2, \ldots, M_N) \in \mathbb{R}^{nN\times nN}$ *such that*

$$\begin{pmatrix} W_1 & \Omega_1 \\ \Omega_1^T & -I_N\otimes\dfrac{B_2+B_2^T}{2}\end{pmatrix} \le 0, \tag{56}$$

where

$$\begin{aligned} W_1 &= I_N\otimes\left(P\Delta+\Delta P-2\lambda I_n+\lambda_0 I_n\right) \\ &\quad -\left(I_N\otimes P\right)Q - Q^T\left(I_N\otimes P\right)+\frac{a}{1-d}M \\ &\quad + a[G\otimes(P\Gamma)]M^{-1}[G\otimes(\Gamma P)], \\ \Omega_1 &= I_N\otimes P-\frac{I_N\otimes B_1^T}{2}. \end{aligned} \tag{57}$$

Proof. Choose the following Lyapunov functional for network (51):

$$V_2(t) = \sum_{i=1}^{N} e_i^T(t)Pe_i(t) + \frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h, \tag{58}$$

where $e(t) = (e_1^T(t), e_2^T(t), \ldots, e_N^T(t))^T$.

According to Ito's lemma, we acquire from (53) and (55)

$$\mathrm{d}V_2(t) = \mathscr{L}V_2(t)\mathrm{d}t + \Phi(t)\mathrm{d}\omega(t). \tag{59}$$

Here

$$\begin{aligned} \mathscr{L}V_2(t) &= 2\sum_{i=1}^{N} e_i^T(t)P\Big[f(z_i(t)) - f(\bar{z}(t)) + a\sum_{j=1}^{N} G_{ij}\Gamma e_j(t-\tau(t)) \\ &\quad + u_i(t) - Q_i e_i(t) - \beta P^{((\alpha-1)/2)}\operatorname{sign}(e_i(t))|e_i(t)|^{\alpha}\Big] \\ &\quad + \sum_{i=1}^{N}\operatorname{trace}[h(z_i(t)) - h(\bar{z}(t))]^T P[h(z_i(t)) - h(\bar{z}(t))] \\ &\quad - 2\beta\sum_{i=1}^{N}\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e_i^T(h)M_i e_i(h)\mathrm{d}h\right)^{((\alpha+1)/2)} \\ &\quad - 2\beta\sum_{i=1}^{N} e_i^T(t)P^{((\alpha+1)/2)}\operatorname{sign}(e_i(t))|e_i(t)|^{\alpha} \\ &\quad + \frac{a}{1-d}\Big[e^T(t)Me(t) \\ &\quad - e^T(t-\tau(t))Me(t-\tau(t))(1-\dot{\tau}(t))\Big], \\ \Phi(t) &= 2\sum_{i=1}^{N} e_i^T(t)P[h(z_i(t)) - h(\bar{z}(t))]. \end{aligned} \tag{60}$$

According to Lemma 3, we can take

$$\begin{aligned} x &= e(t-\tau(t)), \\ y &= [G\otimes(\Gamma P)]e(t). \end{aligned} \tag{61}$$

It is not difficult to obtain

$$\begin{aligned} &2ae^T(t)[G\otimes(P\Gamma)]e(t-\tau(t)) \\ &\quad \le ae^T(t)[G\otimes(P\Gamma)]M^{-1}[G\otimes(\Gamma P)]e(t) \\ &\quad + ae^T(t-\tau(t))Me(t-\tau(t)). \end{aligned} \tag{62}$$

From the above, one has

$$\begin{aligned} \mathscr{L}V_2(t) &\le e^T(t)\big[I_N\otimes\left(P\Delta+\Delta P-2\lambda I_n+\lambda_0 I_n\right) \\ &\quad -\left(I_N\otimes P\right)Q - Q^T\left(I_N\otimes P\right)\big]e(t) \\ &\quad + 2ae^T(t)[G\otimes(P\Gamma)]e(t-\tau(t)) \\ &\quad - 2\beta\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right)^{((\alpha+1)/2)} \\ &\quad - 2\beta\left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)} \\ &\quad - ae^T(t-\tau(t))Me(t-\tau(t)) \\ &\quad + \frac{a}{1-d}e^T(t)Me(t) + 2e^T(t)\left(I_N\otimes P\right)u(t) \end{aligned}$$

$$
\begin{aligned}
&\le e^T(t)\Big[I_N \otimes (P\Delta + \Delta P - 2\lambda I_n + \lambda_0 I_n) \\
&- (I_N \otimes P)Q - Q^T(I_N \otimes P) + \frac{a}{1-d} M \\
&+ a[G \otimes (P\Gamma)]M^{-1}[G \otimes (\Gamma P)]\Big] e(t) \\
&- 2\beta\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right)^{((\alpha+1)/2)} \\
&- 2\beta\left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)} \\
&+ 2e^T(t)(I_N \otimes P)u(t).
\end{aligned} \tag{63}
$$

Thus,

$$
\begin{aligned}
&\mathscr{L}V_2(t) - u^T(t)y(t) \\
&\le \zeta^T(t)\begin{pmatrix} W_1 & \Omega_1 \\ \Omega_1^T & -I_N \otimes \dfrac{B_2 + B_2^T}{2} \end{pmatrix}\zeta(t) \\
&- 2\beta\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right)^{((\alpha+1)/2)} \\
&- 2\beta\left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)} \\
&\le -2\beta\left\{\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right)^{((\alpha+1)/2)}\right. \\
&\left. + \left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)}\right\} \\
&\le -2\beta\left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right. \\
&\left. + \frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right)^{((\alpha+1)/2)} \\
&= -2\beta V_2^{((\alpha+1)/2)}(t),
\end{aligned} \tag{64}
$$

where $u(t) = (u_1^T(t), u_2^T(t), \ldots, u_N^T(t))^T$, $y(t) = (y_1^T(t), y_2^T(t), \ldots, y_N^T(t))^T$, $\zeta(t) = (e^T(t), u^T(t))^T$.

Taking the mathematical expectation on (59), we can obtain

$$
E\{u^T(t)y(t)\} \ge \frac{E\{\mathrm{d}V_2(t)\}}{\mathrm{d}t} + 2\beta E\{V_2^{((\alpha+1)/2)}(t)\}. \tag{65}
$$

Consequently, network model (53) is finite-time passive in the sense of expectation under controller (55). □

Theorem 6. *Under assumptions (H1) and (H2), network model (53) is finite-time input strictly passive in the sense of expectation under controller (55) if there exist matrices $Q = diag(Q_1, Q_2, \ldots, Q_N) \in \mathbb{R}^{nN\times nN}$, $0 < M = diag(M_1, M_2, \ldots, M_N) \in \mathbb{R}^{nN\times nN}$ and a positive real number γ_3 such that*

$$
\begin{pmatrix} W_1 & \Omega_1 \\ \Omega_1^T & \gamma_3 I_{nN} - I_N \otimes \dfrac{B_2 + B_2^T}{2} \end{pmatrix} \le 0, \tag{66}
$$

where W_1, Ω_1 have the same meanings as in Theorem 5.

Proof. We also select the same $V_2(t)$ as (58) for network (53). By (64), we get

$$
\begin{aligned}
&\mathscr{L}V_2(t) - u^T(t)y(t) + \gamma_3 u^T(t)u(t) \\
&\le \zeta^T(t)\begin{pmatrix} W_1 & \Omega_1 \\ \Omega_1^T & \gamma_3 I_{nN} - I_N \otimes \dfrac{A_2 + A_2^T}{2} \end{pmatrix}\zeta(t) \\
&- 2\beta\left\{\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right)^{((\alpha+1)/2)}\right. \\
&\left. + \left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)}\right\} \\
&\le -2\beta\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right. \\
&\left. + \sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)} = -2\beta V_2^{((\alpha+1)/2)}(t).
\end{aligned} \tag{67}
$$

Taking the mathematical expectation on (59), we can obtain

$$
\begin{aligned}
&E\{u^T(t)y(t)\} - \gamma_3 E\{u^T(t)u(t)\} \\
&\ge \frac{E\{dV_2(t)\}}{dt} + 2\beta E\{V_2^{((\alpha+1)/2)}(t)\}.
\end{aligned} \tag{68}
$$

Therefore, network (53) is finite-time input strictly passive in the sense of expectation under controller (55). □

Theorem 7. *Under assumptions (H1) and (H2), network model (53) is finite-time output strictly passive in the sense of expectation under controller (55) if there exist matrices $Q = diag(Q_1, Q_2, \ldots, Q_N) \in \mathbb{R}^{nN\times nN}$, $0 < M = diag(M_1, M_2, \ldots, M_N) \in \mathbb{R}^{nN\times nN}$ and a positive real number γ_4 such that*

$$\begin{pmatrix} W_2 & \Omega_2 \\ \Omega_2^T & W_3 \end{pmatrix} \leq 0, \tag{69}$$

where

$$\begin{aligned} W_2 &= W_1 + \gamma_4\left(I_N \otimes B_1^T\right)\left(I_N \otimes B_1\right), \\ \Omega_2 &= \Omega_1 + \gamma_4\left(I_N \otimes B_1^T\right)\left(I_N \otimes B_2\right), \\ W_3 &= \gamma_4\left(I_N \otimes B_2^T\right)\left(I_N \otimes B_2\right) \\ &\quad - I_N \otimes \frac{B_2 + B_2^T}{2}, \end{aligned} \tag{70}$$

W_1, Ω_1 have the same meanings as in Theorem 5.

Proof. Select the same $V_2(t)$ as (58) for network (53). By (37) and (64), we get

$$\begin{aligned} &\mathscr{L}V_2(t) - u^T(t)y(t) + \gamma_4 y^T(t)y(t) \\ &\leq \zeta^T(t)\begin{pmatrix} W_2 & \Omega_2 \\ \Omega_2^T & W_3 \end{pmatrix}\zeta(t) \\ &\quad - 2\beta\left[\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right)^{((\alpha+1)/2)}\right. \\ &\quad \left. + \left(\sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)}\right] \\ &\leq -2\beta\left(\frac{a}{1-d}\int_{t-\tau(t)}^{t} e^T(h)Me(h)\mathrm{d}h\right. \\ &\quad \left. + \sum_{i=1}^{N} e_i^T(t)Pe_i(t)\right)^{((\alpha+1)/2)} \\ &= -2\beta V_2^{((\alpha+1)/2)}(t). \end{aligned} \tag{71}$$

Taking the mathematical expectation on (59), we can obtain

$$\begin{aligned} &E\left\{u^T(t)y(t)\right\} - \gamma_4 E\left\{y^T(t)y(t)\right\} \\ &\geq \frac{E\{\mathrm{d}V_2(t)\}}{\mathrm{d}t} + 2\beta E\left\{V_2^{((\alpha+1)/2)}(t)\right\}. \end{aligned} \tag{72}$$

Therefore, network (53) is finite-time output strictly passive in the sense of expectation under controller (55). □

4.3. Finite-Time Synchronization

Theorem 8. *Assume that a continuous, positive-definite function $\widehat{V}(t)$ satisfies the following inequality:*

$$\varphi_2\left(E\|e(t)\|_2\right) \leq \widehat{V}(t), \tag{73}$$

where φ_2: $[0, +\infty) \longrightarrow [0, +\infty)$ is continuous and strictly monotonically increasing function and $\varphi_2(s)$ is positive for $s > 0$ with $\varphi_2(0) = 0$. If network (51) is finite-time passive (finite-time input strictly passive, finite-time output strictly passive) in the sense of expectation with respect to $\widehat{V}(t)$, then SCCN (53) is finite-time synchronized in the sense of expectation under controller (55).

Here we omit the proof of the theorem. The readers can refer to the proof of Theorem 4.

5. Numerical Examples

Example 1. The following SCCNs are discussed:

$$\begin{aligned} \mathrm{d}z_i(t) = &\left[f\left(z_i(t)\right) + 0.7\sum_{j=1}^{6} G_{ij}\Gamma z_j(t) + u_i(t)\right. \\ &\left. + v_i(t)\right]\mathrm{d}t + h\left(z_i(t)\right)\mathrm{d}\omega(t), \quad i = 1, 2, \ldots, 6, \end{aligned} \tag{74}$$

where $f_s(\varsigma) = (1/4)(|\varsigma + 1| - |\varsigma - 1|), s = 1, 2, 3, \Gamma = \mathrm{diag}(0.15, 0.05, 0.25)$, $h(z_i(t)) = \mathrm{diag}(0.2, 0.4, 0.2)$,

$$G = \begin{pmatrix} -0.4 & 0.1 & 0 & 0.1 & 0.2 & 0 \\ 0.1 & -0.4 & 0 & 0.2 & 0.1 & 0 \\ 0 & 0 & -0.5 & 0.2 & 0.1 & 0.2 \\ 0.1 & 0.2 & 0.2 & -0.6 & 0 & 0.1 \\ 0.2 & 0.1 & 0.1 & 0 & -0.5 & 0.1 \\ 0 & 0 & 0.2 & 0.1 & 0.1 & -0.4 \end{pmatrix}. \tag{75}$$

Obviously, $H(1)$ holds under the condition that $P = I_3, \lambda = 0.8$, and $\Delta = \mathrm{diag}(0.15, 0.14, 0.12)$. Choose $A_2 = \mathrm{diag}(0.6, 0.8, 0.9)$ and

$$A_1 = \begin{pmatrix} 0.3 & 0.2 & 0.1 \\ 0.4 & 0.3 & 0.3 \\ 0.6 & 0.1 & 0.4 \end{pmatrix}. \tag{76}$$

Take $Q = \mathrm{diag}(0.2I_3, 0.6I_3, 0.7I_3, 0.5I_3, 0.3I_3, 0.4I_3)$. According to Theorem 1, the SCCNs (74) can realize finite-time passivity in the sense of expectation under controller (17). Then, we can easily find the parameters $\gamma_1 = 0.0163$ and $\gamma_2 = 0.0300$ satisfying the condition of Theorems 2 and 3. The simulation results are shown in Figures 1 and 2.

By Theorem 4, network (74) under finite-time output strictly passive can achieve finite-time synchronization. Figure 3 shows the simulation results.

Example 2. The following SCCNs with time-varying delay are discussed:

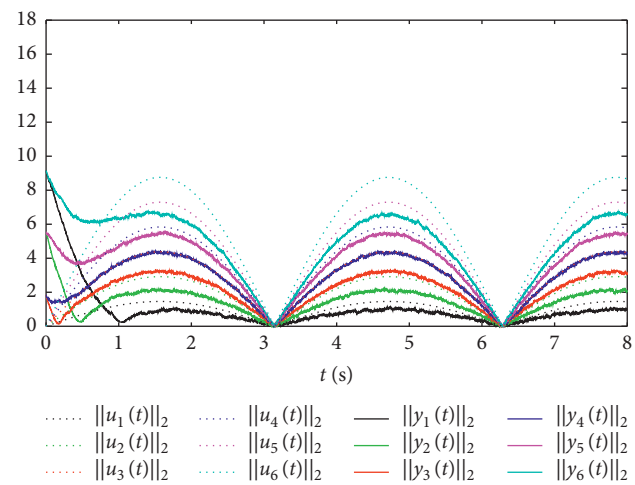


Figure 1: The norms of the input and output vectors $\|u_i(t)\|_2$, $\|y_i(t)\|_2$, $i = 1, 2, \ldots, 6$.

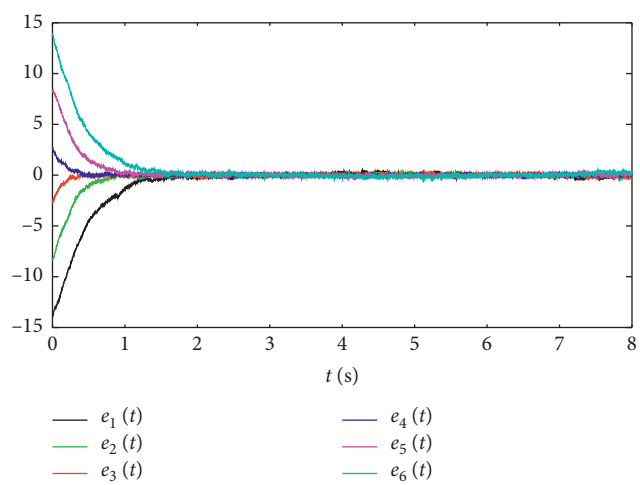


Figure 2: The error vectors $e_i(t)$, $i = 1, 2, \ldots, 6$.

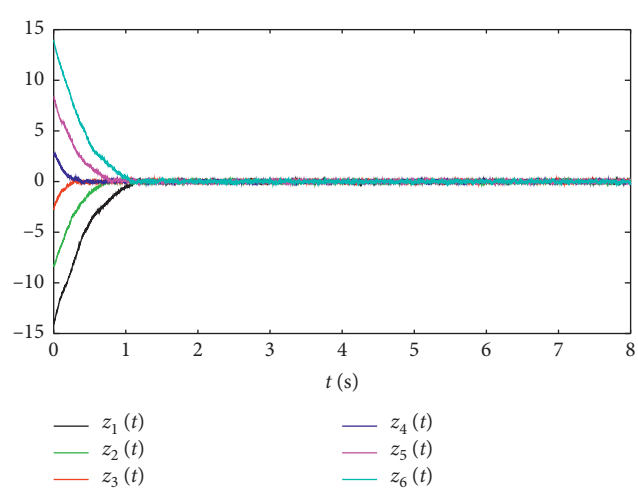


Figure 3: The state vectors $z_i(t)$, $i = 1, 2, \ldots, 6$.

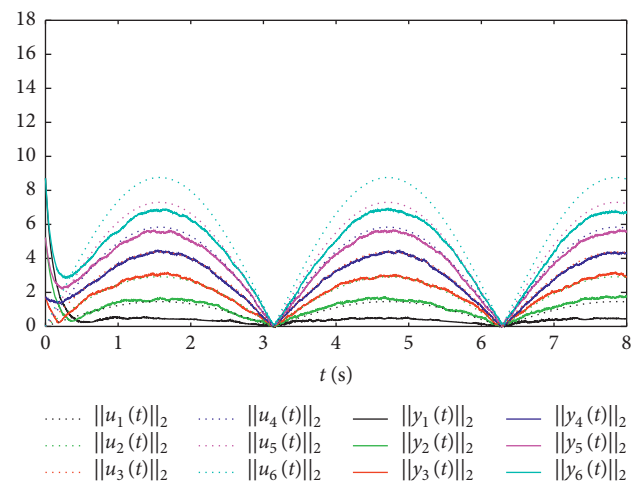


Figure 4: The norms of the input and output vectors $\|u_i(t)\|_2$, $\|y_i(t)\|_2$, $i = 1, 2, \ldots, 6$.

$$\begin{aligned} dz_i(t) = \Big[& f(z_i(t)) + 0.8\sum_{j=1}^{6} G_{ij}\Gamma z_j(t-\tau(t)) \\ & + u_i(t) + v_i(t)\Big]dt + h(z_i(t))d\omega(t), \quad i = 1, 2, \ldots, 6, \end{aligned} \tag{77}$$

where $f_s(\varsigma) = (1/4)(|\varsigma + 1| - |\varsigma - 1|), s = 1, 2, 3$, $\Gamma = \mathrm{diag}(0.15, 0.15, 0.15)$, $h(z_i(t)) = \mathrm{diag}(0.1, 0.2, 0.2)$. Taking $\tau(t) = 0.5 - 0.5e^{-t}$, we can get $d = 0.5$. The matrix G is chosen as

$$G = \begin{pmatrix} -0.8 & 0.1 & 0.2 & 0.1 & 0.2 & 0.2 \\ 0.1 & -0.5 & 0 & 0.2 & 0.1 & 0.1 \\ 0.2 & 0 & -0.7 & 0.2 & 0.1 & 0.2 \\ 0.1 & 0.2 & 0.2 & -0.6 & 0 & 0.1 \\ 0.2 & 0.1 & 0.1 & 0 & -0.5 & 0.1 \\ 0.2 & 0.1 & 0.2 & 0.1 & 0.1 & -0.7 \end{pmatrix}. \tag{78}$$

Obviously, $H(1)$ holds under the condition that $P = I_3, \lambda = 0.7$, and $\Delta = \mathrm{diag}(0.18, 0.16, 0.15)$. Choose $B_2 = \mathrm{diag}(0.6, 0.8, 0.9)$ and

$$B_1 = \begin{pmatrix} 0.6 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.1 \\ 0.3 & 0.3 & 0.4 \end{pmatrix}. \tag{79}$$

Take $Q = \mathrm{diag}(0.5I_3, 0.6I_3, 0.7I_3, 0.5I_3, 0.7I_3, 0.8I_3)$, $M = I_6 \otimes \mathrm{diag}(0.1, 0.2, 0.4)$. According to Theorem 5, the SCCNs (77) can realize finite-time passivity under controller (55). Then, we can easily find the parameters $\gamma_3 = 0.0132$ and $\gamma_4 = 0.0369$ satisfying the condition of Theorems 6 and 7. The simulation results are shown in Figures 4 and 5.

By Theorem 8, network (77) under finite-time output strictly passive can achieve finite-time synchronization. Figure 6 shows the simulation results.

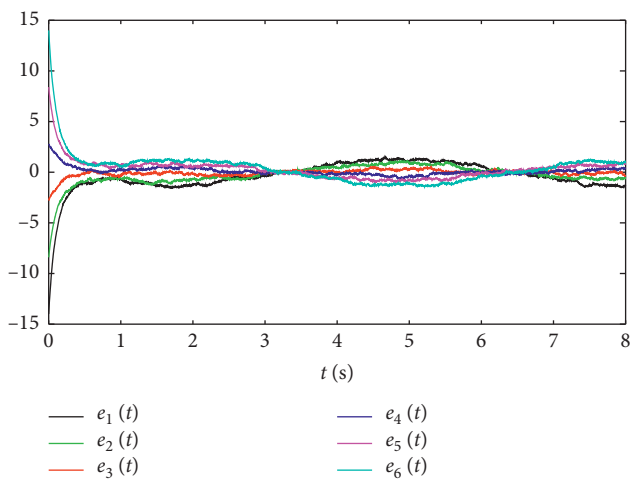


Figure 5: The error vectors $e_i(t)$, $i = 1, 2, \ldots, 6$.

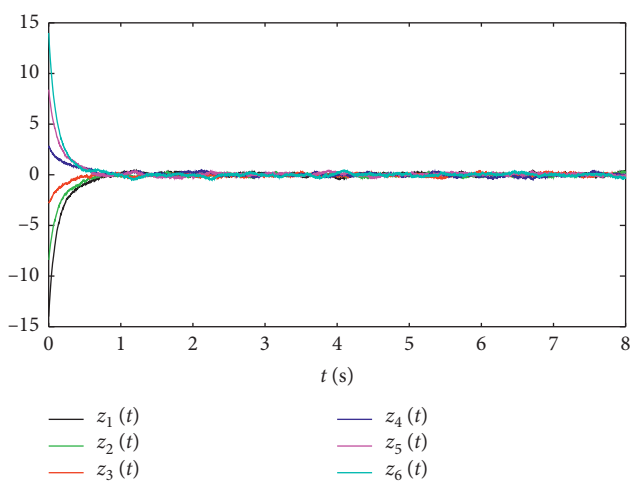


Figure 6: The state vectors $z_i(t)$, $i = 1, 2, \ldots, 6$.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under grant nos. 61773285 and 11871371.

References

[1] J. Ruths and D. Ruths, "Control profiles of complex networks," *Science*, vol. 343, no. 6177, pp. 1373–1376, 2014.

[2] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex net- works," *Science*, vol. 353, no. 6295, pp. 163–165, 2016.

[3] J. Gao, B. Barzel, and A. Barabsi, "Universal resilience patterns in complex networks," *Nature*, vol. 530, no. 7590, pp. 307–312, 2016.

[4] J. Liang, Z. Wang, and X. Liu, "Exponential synchronization of stochastic delayed discrete time complex networks," *Nonlinear Dynamics*, vol. 53, pp. 153–165, 2008.

[5] C. Li, L. Chen, and K. Aihara, "Stochastic synchronization of genetic oscillator networks," *BMC Systems Biology*, vol. 1, pp. 1–6, 2007.

[6] Y. Wang, Z. Wang, and J. Liang, "A delay fractioning approach to global synchronization of delayed complex networks with stochastic disturbances," *Physics Letters A*, vol. 372, no. 9, pp. 6066–6073, 2008.

[7] Z. Wang, Y. Wang, and Y. Liu, "Global synchronization for discrete time stochastic complex networks with randomly occurred nonlinearities and mixed time delays," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 11–26, 2010.

[8] H. Li and D. Yue, "Synchronization of Markovian jumping stochastic complex networks with distributed time delays and probabilistic interval discrete time varying delays," *Journal of Physics A: Mathematical and Theoretical*, vol. 43, no. 10, 2010.

[9] X. Lou and B. Cui, "Passivity analysis of integro-differential neural networks with time-varying delays," *Neurocomputing*, vol. 70, pp. 1071–1078, 2007.

[10] J. Yao, H. O. Wang, Z.-H. Guan, and W. Xu, "Passive stability and synchronization of complex spatio-temporal switching networks with time delays," *Automatica*, vol. 45, pp. 1721–1728, 2009.

[11] J. L. Wang, H. N. Wu, and L. Guo, "Passivity and stability analysis of reaction-diffusion neural networks with dirichlet boundary conditions," *IEEE Transactions on Neural Networks*, vol. 22, pp. 2105–2116, 2011.

[12] X. W. Yin, X. N. Song, and M. Wang, "Passive fuzzy control design for a class of nonlinear distributed parameter systems with time-varying Delay," *International Journal of Control, Automation and Systems*, vol. 18, no. 4, pp. 911–921, 2020.

[13] Z. G. Wu, P. Shi, and J. Chu, "Passivity analysis for discrete-time stochastic Markovian jump neural networks with mixed time delays," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1566–1575, 2011.

[14] Q. Song, J. Liang, and Z. Wang, "Passivity analysis of discrete-time stochastic neural networks with time-varying delays," *Neurocomputing*, vol. 72, pp. 1782–1788, 2009.

[15] H. Y. Du, W. S. Sun, G. Hu, and L. H. Qi, "Function projective synchronization of two fractional-order complex dynamical networks," *Acta Automatica Sinica*, vol. 42, no. 2, pp. 226–234, 2016.

[16] V. Andrieu, B. Jayawardhana, and S. Tarbouriech, "Some results on exponential synchronization of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1213–1219, 2018.

[17] D. Zhang, Y. Shen, and J. Mei, "Finite-time synchronization of multi-layer nonlinear coupled complex networks via intermittent feedback control," *Neurocomputing*, vol. 225, pp. 129–138, 2017.

[18] X. Lu, X. Zhang, and Q. Liu, "Finite-time synchronization of nonlinear complex dynamical networks on time scales via pinning impulsive control," *Neurocomputing*, vol. 275, pp. 2104–2110, 2018.

[19] M. Tan, X. Li, and Y. Liu, "Finite-time stability and synchronization of the coupled switched neural networks with nodes of different dimensions," *Neural Processing Letters*, vol. 49, no. 1, pp. 285–303, 2019.

[20] X. Yang and J. Lu, "Finite-time synchronization of coupled networks with Markovian topology and impulsive effects," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2256–2261, 2016.

[21] E. Moulay and M. Dambrine, "Finite-time stability and stabilization of time-delay systems," *System & Control Letter*, vol. 57, no. 7, pp. 561–566, 2008.

[22] J. L. Wang, M. Xu, H. N. Wu, and T. W. Huang, "Finite-time passivity of coupled neural networks with multiple weights," *IEEE Transcations on Network Science and Engineering*, vol. 5, no. 3, pp. 184–197, 2018.

[23] Y. Tang, "Terminal sliding mode control for rigid robots," *Automatica*, vol. 34, pp. 51–56, 1998.

[24] X. Huang, W. Lin, and B. Yang, "Global finite-time stabilization of a class of uncertain nonlinear systems," *Automatica*, vol. 41, pp. 881–888, 2005.

[25] J. L. Wang, H. N. Wu, and S. Y. Ren, *Passivity of Complex Dynamical Networks*, Springer, Berlin, Germany, 2021.

[26] W. L. Lu and T. P. Chen, "New approach to synchronization analysis of linearly coupled ordinary differential systems," *Physica D*, vol. 213, no. 2, pp. 214–230, 2006.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 9998185, 11 pages
https://doi.org/10.1155/2021/9998185

Research Article

Spectral-Spatial Hyperspectral Image Semisupervised Classification by Fusing Maximum Noise Fraction and Adaptive Random Multigraphs

Eryang Chen,[1,2,3,4] Ruichun Chang,[4,5] Kaibo Shi,[2] Ansheng Ye,[1,2] Fang Miao,[3] Jianghong Yuan,[6] Ke Guo,[4] Youhua Wei,[4] and Yiping Li[1,4]

[1]*College of Geophysics, Chengdu University of Technology, Chengdu 610059, China*
[2]*School of Electronic Information and Eletical Engineering, Chengdu University, Chengdu 610106, China*
[3]*Key Laboratory of Pattern Recognition and Intelligent Information Processing of Sichuan, Chengdu University, Chengdu 610106, China*
[4]*Geomathematics Key Laboratory of Sichuan Province, Chengdu University of Technology, Chengdu 610059, China*
[5]*Digital Hu Line Research Institute, Chengdu University of Technology, Chengdu 610059, China*
[6]*School of Intelligent Engineering, Sichuan Changjiang Vocational College, Chengdu 610106, China*

Correspondence should be addressed to Ruichun Chang; 2293721870@qq.com and Kaibo Shi; skbs111@163.com

Received 12 March 2021; Accepted 15 June 2021; Published 24 June 2021

Academic Editor: Zi-Peng Wang

Hyperspectral images (HSIs) contain large amounts of spectral and spatial information, and this provides the possibility for ground object classification. However, when using the traditional method, achieving a satisfactory classification result is difficult because of the insufficient labeling of samples in the training set. In addition, parameter adjustment during HSI classification is time-consuming. This paper proposes a novel fusion method based on the maximum noise fraction (MNF) and adaptive random multigraphs for HSI classification. Considering the overall spectrum of the object and the correlation of adjacent bands, the MNF was utilized to reduce the spectral dimension. Next, a multiscale local binary pattern (LBP) analysis was performed on the MNF dimension-reduced data to extract the spatial features of different scales. The obtained multiscale spatial features were then stacked with the MNF dimension-reduced spectral features to form multiscale spectral-spatial features (SSFs), which were sent into the RMG for HSI classification. Optimal performance was obtained by fusion. For all three real datasets, our method achieved competitive results with only 10 training samples. More importantly, the classification parameters corresponding to different hyperspectral data can be automatically optimized using our method.

1. Introduction

Due to the advancements in remote sensing technology, hyperspectral images (HSIs) are containing increasing spectral and spatial information (SSI), resulting in their extensive use in domains, such as forest inventory [1], urban area monitoring [2], road extraction [3], geological surveys [4], precision agriculture [5], environmental protection [6], military applications [7], hydrocarbon detection [8], oil reservoir exploration [9], and lake sediment analysis [10]. HSI classification is a crucial research topic related to these applications. In contrast to those of Synthetic Aperture Radar (SAR) [11] or RGB images [12], the two main challenges associated with HSI classification are the high dimensionality of the dataset and the redundancy of spectral information.

Many approaches for HSI dimension reduction have been proposed. The most frequently used methods are principal component analysis (PCA) [13, 14], independent component analysis (ICA) [15], maximum noise fraction (MNF) [16], linear discriminant analysis [17], and the deep learning approach [18]. Uddin et al. [13] proposed an information-theoretic normalized-based PCA for HSI classification. Fu et al. [14] proposed a segmented PCA-based algorithm for HSI classification.

However, because the noise in hyperspectral data can easily mask subtle hyperspectral features, careful noise removal is required to extract useful information. This phenomenon is problematic for PCA, which maximizes the variance of the orthogonal set of the projection sample and vector. Unlike PCA, MNF aims to maximize the signal-to-noise ratio (SNR) rather than the variance. MNF is worth considering in HSI classification because removing the noise is effective during dimensionality reduction.

HSI classification can be significantly improved by employing an appropriate object-based [19], spectral-spatial fusion-based approach [20], decision fusion-based method [21], or deep learning-based method [22]. However, two challenges are unavoidable for most researchers. On the one hand, the traditional depth learning-based approaches rely heavily on a large amount of labeled data to achieve competitive results, whereas HSI annotation is expensive and time-consuming because it requires expert knowledge and skills. On the other hand, many methods require the use of a large number of parameter settings during the experiment, which involves expert experience [23]. HSI classification has been widely studied, and various classification methods have been adopted, namely, support vector machine (SVM) [24], random forests [25], neural networks [26], low-rank representation [27], sparse representation [28], deep learning methods [29, 30], and meta-learning methods [31]. Xu et al. [29] directly used the random patches extracted from the HSI image as the convolution kernel, without any training, to improve the classification efficiency. Yin et al. [30] proposed a CapsNet-based alternative data-driven HSI classification model. The aforementioned results demonstrate that the proposed method can achieve ideal results when the training samples are sufficient.

To integrate the spatial information, researchers have proposed many spectral-spatial feature extraction (FE) techniques, such as the extinction profile (EP) [32] and local binary patterns (LBP) [33]. In contrast with the EP features, LBP features facilitate the mining of the HSI texture information, such as global contrast information and texture depth [34]. In this study, we adopted the LBP features as spatial features.

However, when the labeled samples in the training set are insufficient, the classification accuracy achieved by the traditional methods significantly reduces [35] because of the so-called Hughes effect or the curse of dimensionality [36]. Moreover, many methods require a series of manual parameter settings. For instance, to extract spatial features, researchers must select an appropriate window size for the neighborhood. The selection is time-consuming [37] and requires expertise [38]. Since 2010, ensemble learning methods for HSI classification have received significant attention because of their dependence on limited training samples [39]. Many methods have been developed, such as support vector machines [40], boosting [41], segmentation-based methods [19], unsupervised methods [42], and semisupervised methods [43]. However, the graph-based semisupervised method is rarely considered in HSI classification [44, 45].

Motivated by the aforementioned discussions, by combining the SSI, we proposed a new spectral-spatial HSI semisupervised classifier based on MNF and adaptive random multigraphs (SS-MNF-ARMG). The primary contributions of this study are as follows:

(1) A novel spectral-spatial HSI semisupervised classification framework was developed. Because of the adaptive properties, the optimal parameters can be determined without artificial auxiliaries.

(2) By introducing the MNF, the noise in the HSI can be removed more efficiently during dimensionality reduction. On the basis of dimension-reduced HSI, the SSI is combined, which can degrade the curse of dimensionality.

(3) In contrast with several studies, SS-MNF-ARMG can achieve competitive performance for three real HSI datasets while leveraging tiny labeled samples, which is further improved by introducing RMG in a new mode.

2. Relevant Work

2.1. MNF. Let X be the HSI data, and S and N are the signal and noise part of X, respectively. The goal of MNF is to seek out a linear transformation matrix W to maximize the SNR of the transformed data. Assume that S and N are uncorrelated; then X can be represented as [43]

$$X = S + N. \tag{1}$$

And, the covariance matrix (CM) of X can be obtained by

$$\operatorname{cov}(X) = \sum_X = \sum_S + \sum_N, \tag{2}$$

where $\sum_S$ and $\sum_N$ are the CMs of S and N, respectively. The MNF transform can be expressed as

$$Y = W^T X, \tag{3}$$

where Y is the MNF result of X, W are the eigenvectors associated with the L largest eigenvalues of $\sum_N^{-1} \sum_X$, and L is the number of MNF principal components.

Then the SNR of each $y_i \in Y$ can be described as

$$\frac{\operatorname{cov}\left\{w_i^T S\right\}}{\operatorname{cov}\left\{w_i^T N\right\}} = \frac{w_i^T \sum_S w_i}{w_i^T \sum_X w_i}, \tag{4}$$

where $\operatorname{cov}\{\cdot\}$ computes the variance and w_i is the ith component in W. Then we can obtain W by solving the following problem:

$$\arg\max_W \frac{W^T \sum_S W}{W^T \sum_N W} = \arg\max_W \frac{W^T \sum_X W}{W^T \sum_N W} - 1. \tag{5}$$

2.2. LBP. For each pixel in HSI, a P-bit binary code and a new matrix with the new value (binary to decimal value) are generated by thresholding adjacent pixel values. The LBP operator can be calculated by

$$\mathrm{LBP}_{i,r}(N_c) = \sum_{i=0}^{P-1} g(N_i - N_c)2^i, \tag{6}$$

where P is the number of neighbors represented on a circle with the radius r, and N_c and N_i, respectively, represent the gray level intensity values of the center and the ith neighbour. The binary threshold function $g(x)$ is described as

$$g(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases} \tag{7}$$

Take the 10th band of Indian Pines HSI as an example; the procedure of LBP is shown in Figure 1. As shown in Figure 1, for a given center pixel in a 3×3 window, binary labels ("0" or "1") are assigned to adjacent pixels according to whether the gray value of the center pixel is large. Starting at the top left corner, all binary codes are joined clockwise to produce an 8-bit binary number. The resulting binary number is called LBP code. The results show that LBP algorithm has significant rotation invariance and gray invariance [46], and it can be effectively applied to HSI classification [47].

2.3. RMG. Given the HSI dataset $D = X_{lab} \cup X_{ulab}$ is comprised of l labeled data $X_{\mathrm{lab}} = \{(x_i)\}_{i=1}^{l}$ and u unlabeled data $X_{ulab} = \{(x_i)\}_{i=l+1}^{l+u}$, then a weighted graph can be obtained. The vertices in the graph consist of X_{lab} and X_{ulab}. Weighted edges, which can be defined as a matrix $W = \mathbb{R}^{(l+u)\times(l+u)}$, represent the similarities between associated nodes. For a c-class classification problem, it can be defined as a quadratic optimization problem [43]:

$$\min_F tr\left((F-Y)^T C (F-Y)\right) + tr\left(F^T L F\right), \tag{8}$$

where $tr(\cdot)$is the trace function and $F \in \mathbb{R}^{(l+u)\times c}$ is the prediction matrix to be learned. The indicator vector $y_i \in Y = (y_1, y_2, \ldots, y_l, 0, \ldots, 0)^T \in \mathbb{R}^{(l+u)\times c}$ is the label vector corresponding to x_i and 0 is the 0 vector. In addition, we can obtain y_{ij} by the following equation:

$$y_{ij} = \begin{cases} 1, & \text{if } x_i \text{ belongs to the } j\text{th class}, \\ 0, & \text{otherwise}. \end{cases} \tag{9}$$

Then each x_i can be classified to the jth class if F_{ij} is the largest one in the ith row of F which can be described as $y_i = \arg\max F_{ij}, j = 1, \ldots, c$. $C \in \mathbb{R}^{(l+u)\times(l+u)}$ is a diagonal matrix, and its element c_i can be calculated as

$$c_i = \begin{cases} \alpha_l, & 1 \leq i \leq l, \alpha_l > 0, \\ \alpha_u, & l+1 \leq i \leq l+u, \alpha_u \geq 0, \end{cases} \tag{10}$$

where α_l and α_u are two parameters.

The popular choice for$L$is the graph Laplacian [48], which is defined as

$$L = D - W, \tag{11}$$

where W is the weight matrix of the graph, which can be formulated by Gaussian kernel as

$$w_{ij} = e^{-\left(\|x_i - y_i\|^2/2\sigma^2\right)} \tag{12}$$

where σ is the kernel width parameter to be adjusted. And the diagonal matrix D is the row sum of W.

However, it is difficult to discover the neighborhood structure inherent in the graph and learn the proper compact representation automatically. To solve this problem, researchers have proposed the anchor graph algorithm (AGA) [49] and multiview anchor graph algorithm [50]. The AGA extrapolates the Laplacian eigenvector of the graph to the eigenfunction, allowing constant time hashing of new data points. Then the hierarchical threshold learning method is used to make each feature function generate more than 1 bit to improve the search accuracy. And the label prediction function in the AGA can be expressed as

$$f(x_i) = \sum_{j=1}^{m} P_{ij} f(a_j), \tag{13}$$

where$P_{ij}$is the data-adaptive weight, and each a_j in $A = \{a_j\}_{j=1}^{m}$ is an anchor point. By this formula, the solution space of the unknown labels can be reduced from a larger space to a smaller one. The centers of the K-means cluster are selected as anchor points because these centers have a powerful representation that covers the entire dataset. In this paper, we use Local Anchor Embedding (LAE) [49] to calculate the anchor points. Figure 2 shows the flowchart of RMG.

3. Proposed Method

The SS-MNF-ARMG framework is shown in Figure 3. Because of their adaptive properties, the optimal parameters can be determined without artificial auxiliaries. The framework comprises three main modules:

(1) Preprocessing the HSI image by applying MNF, the noise in the HSI can be removed effectively during dimension reduction. This result avoids the problem of the dimension.

(2) Through LBP, spatial vectors corresponding to different neighborhood regions are obtained. These spatial vectors are stacked with the spectral vector, respectively, and a series of spectral-spatial stacked feature information can be obtained.

(3) For classification and decision fusion, a set of necessary parameters for the RMG is established and the spectral-spatial feature information is integrated into the RMG. Next, a set of classification results with different accuracies are obtained, based on which the optimal classification results are obtained through decision fusion. In addition, by injecting randomness into the graph in the RMG, overfitting due to the limited training sample can be avoided.

The proposed SS-MNF-ARMG algorithm is summarized in Algorithm 1.

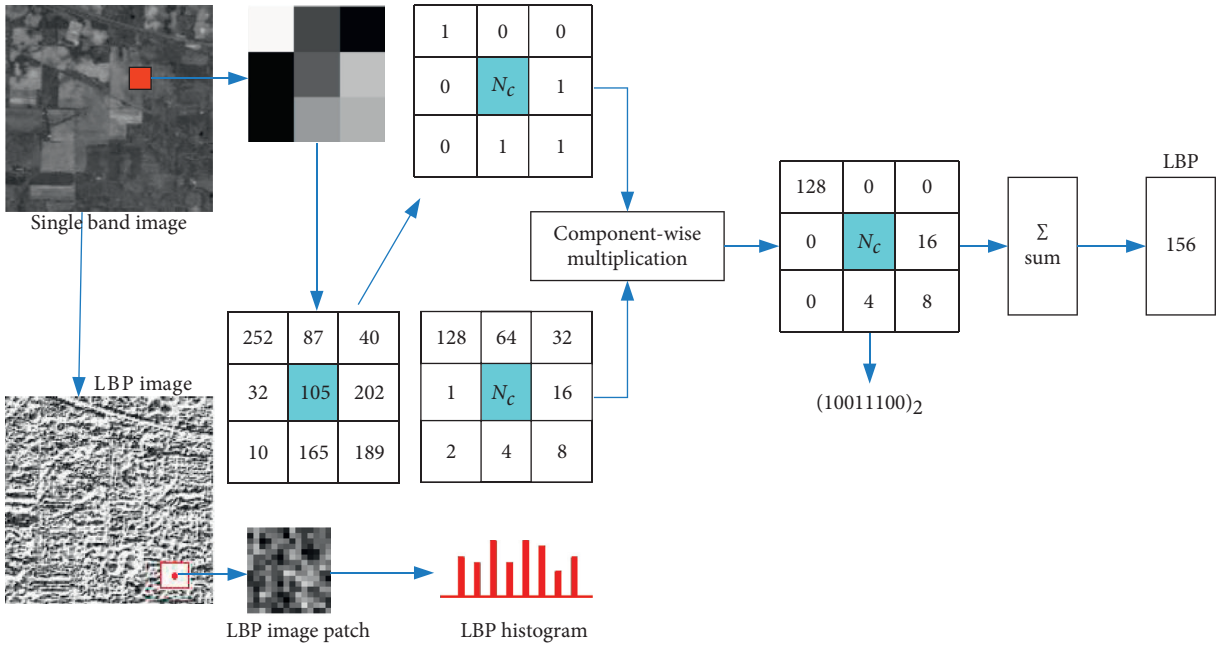


Figure 1: Implementation of LBP FE.

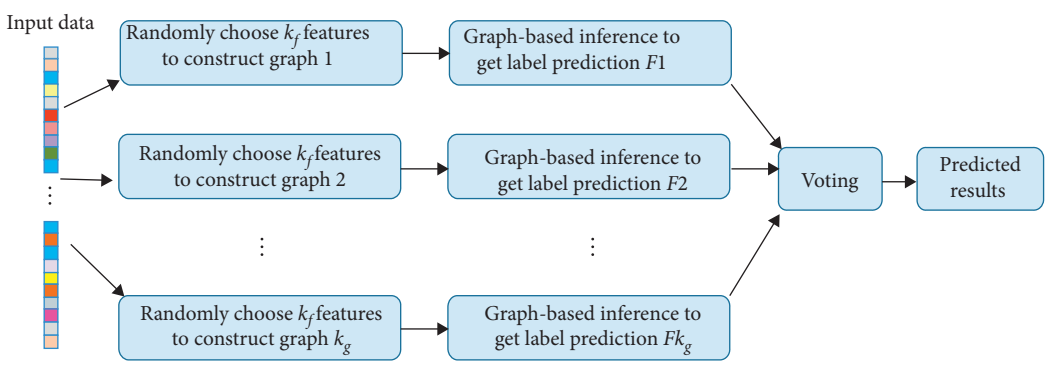


Figure 2: Flowchart of the RMG.

4. Experimental Results and Analysis

4.1. Experimental Datasets. Three hyperspectral datasets were employed to evaluate the performance of the SS-MNF-ARMG.

Indian Pines: this scene was gathered by the AVIRIS sensor over the Indian Pines test site in northwestern Indiana and consists of 145×145 pixels and 224 spectral reflectance bands in the wavelength range 0.4 to 2.5 μm. This scene, which includes 16 different ground truths, contains two-thirds of agriculture and one-third of forest or other natural perennial vegetation. The number of bands was reduced to 200 by removing the 24 water absorption bands.

Pavia University: this scene was acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy. The number of spectral bands was 103 at Pavia University. It is a 610×340 pixel image containing nine different ground objects with a geometric resolution of 1.3 meters.

Salinas: this scene was collected by the 224-band AVIRIS sensor over the agricultural area of Salinas Valley, California, and is characterized by high spatial resolution (3.7 m pixels). After discarding 20 water absorption bands, the size of this data image was 512×217, with 204 bands. Salinas ground truth contains 16 classes, including vegetables, bare soils, and vineyard fields.

4.2. Analysis of Experimental Parameters. Parameters can affect the classification accuracy, such as the number of spectral bands *ban dN um*, the patch size W, the number of sampling points *nr*, the number of graphs kg, and the number of features kf. The adaptive properties of the proposed SS-MNF-ARMG are such that most optimal parameters for W, nr, kg, and kf can be determined without artificial auxiliaries.

Based on the existing result [44], we let $W \in \{15, 19, 23\}$, $\mathrm{nr} \in \{15, 18, 21\}$, $\mathrm{kg} \in \{19, 25, 31\}$, and $\mathrm{kf} \in \{2\sqrt{\dim(V_{\mathrm{spe\_spa}})}, 4\sqrt{\dim(V_{\mathrm{spe\_spa}})}, 8\sqrt{\dim(V_{\mathrm{spe\_spa}})}\}$, where $\dim(V_{\mathrm{spe\_spa}})$ represents the dimensionality of the spectral-spatial vector $V_{\mathrm{spe\_spa}}$.

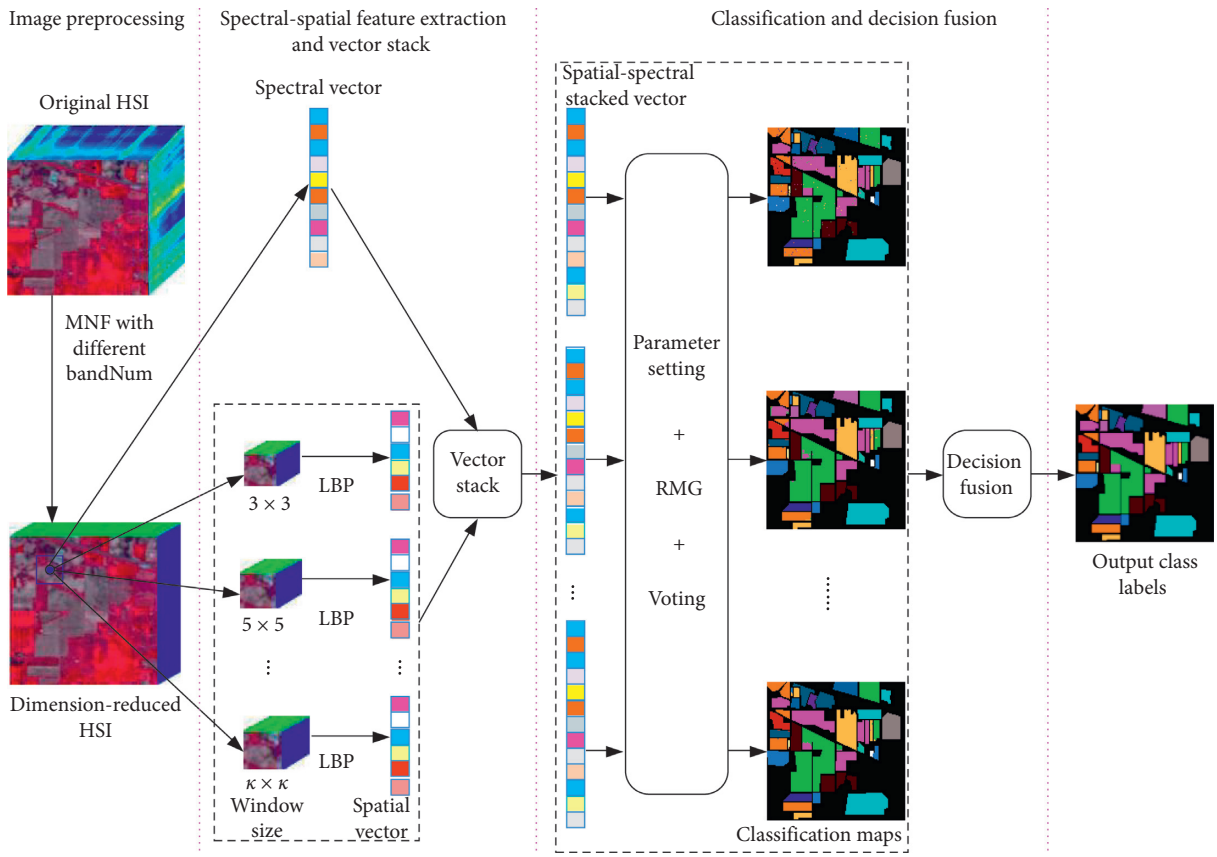


Figure 3: Schematic of the SS-MNF-ARMG algorithm.

Input: the original HSI set is $D \in \mathbb{R}^{x \times y \times h}$; the training set is tra_s; the test set is tes_s; the number of spectra after dimension reduction is $bandNum = [h'_1, h'_2, \ldots, h'_k]$; the patch size for FE is $W = [w_1, w_2, \ldots, w_t]$, where$\{w_i\}_{i=1}^{t}$is odd and $w_i \geq 3$;
Output: the best classification result of all test samples;
(1) **for** each $h'_i \in bandNum$ **do**
(2) Obtain the dimension-reduced HSI $D'_i \in \mathbb{R}^{x \times y \times h'_i}$ by using MNF;
(3) Extract the spectral vector Spe_vec of D'_i;
(4) **for** each $w_j \in W$ **do**
(5) Calculate the spatial vector Spa_vec_j by using LBP;
(6) Obtain the spectral-spatial vector by stacking the Spe_vec and the Spa_vec_j;
(7) Obtain the best $OA_{i,j}$ (overall accuracy) by voting, and the corresponding confusion matrix $ConfMat_{i,j}$;
(8) end for
(9) Obtain the best OA' from $OA_{i,j}$ by decision fusion, and the corresponding confusion matrix $ConfMat'$;
(10) end for

Algorithm 1: SS-MNF-ARMG.

In this study, we set the number of bands from 3 to 35 to evaluate the impact on the three HSI datasets. We conducted the experiment five times, and the optimal experimental results are shown in Figure 4.

It can be observed that, on the Indian Pines dataset and the Salinas scene dataset, the value of overall classification accuracy (OA) shows a trend of steady growth. In other words, the increase in the number of spectral bands number of training has a positive promotion on the classification performance. In particular, on the Pavia University dataset, there is a sharp growth when the number of bands is above 15. In general, the OA of the proposed method is above 78% for all three HSI datasets. Especially in the Salinas scene dataset, the OA of the method is not less than 92% even with a small number of spectral bands, demonstrating the robustness of our method.

4.3. Comparison and Analysis. The proposed SS-MNF-ARMG method was compared with several state-of-the-art spectral-spatial fusion methods, such as Pixon-based classifier [19], PCA-SPCA-2D-SSA [14], R-VCANet [20], RN-

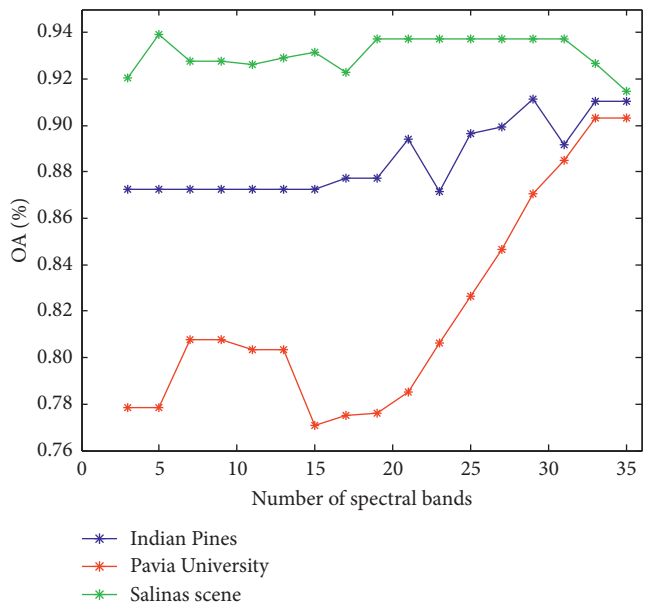


Figure 4: Influence of the spectral band numbers.

Table 1: A comparison of different methods for hyperspectral image classification.

Group	Subgroup	Method	Advantages	Difficulties
Segmentation-based methods	Object-based classification	Pixon-based classifier	The noise pixels in the classification map were removed and the suitable land cover smoothing map was obtained	The spectral characteristics of abnormal pixels are often similar to the background, resulting in the loss of information when the abnormal pixels are removed
Feature fusion	Features stacking	PCA-SPCA-2D-SSA	Combined with appropriate spatial features, the classification efficiency of the algorithm will be higher	Complex structure
	Joint spectral-spatial FE	R-VCANet	The use of high correlation among SSI	Complex structure
	Meta-learning-based classifiers	RN-FSC	Less sensitive to the number of training samples	Limited generalization ability for large-scale hyperspectral datasets
	Deep learning-based classifiers	iCapsNet	Well-initialized shallow layers	Complex structure
		RPNet	FE and classification are carried out under a unified framework	Insufficient training samples will lead to overfitting
Decision fusion	Decision fusion	MBFSDA	Combination of supplemental information and several advanced classifiers	Selection of suitable feature extractor
		Our method	(i) Self-adaptability of parameters (ii) Leveraging tiny labeled samples	Complex structure

FSC [31], iCapsNet [30], RPNet [29], and MBFSDA [21]. A comparison of the above algorithms can be seen in Table 1.

To quantitatively compare the classification performance of the methods shown in Table 1, we used the average classification accuracy (AA), overall classification accuracy (OA), and the kappa coefficient (kappa) to assess the classification performance. To demonstrate the superior performance of the SS-MNF-ARMG with a limited number of training samples, we randomly selected 10 samples from each class as training samples. Tables 2–4 show the ground truth classes and their respective training and testing numbers for the three HSI datasets. Tables 5–7 summarize the experimental results for the three HSI datasets, from which the following conclusions can be drawn:

(1) The results on the Indian Pines dataset show that almost all algorithms are effective. We can observe from Table 5 that the proposed SS-MNF-ARMG achieves a 3.78–28.6% advantage over the other methods in OA. In addition, for the classes that other methods do not recognize accurately, SS-MNF-ARMG can obtain better results, such as objects 1#, 2#, 3#, 5#, 6#, and 7# in Table 5.

(2) The results on the Pavia University dataset demonstrate that our method has advantages over some state-of-the-art methods. As a decision fusion-based algorithm, the proposed SS-MNF-ARMG surpasses MBFSDA by 3.87% in OA. In addition, for the classes that other methods do not recognize accurately, SS-

Table 2: The ground truth classes and their respective training and testing numbers of the Indian Pines datasets (10 classes).

No.	Name of class	Training	Testing
1	Corn-notill	10	1418
2	Corn-mintill	10	820
3	Grass-pasture	10	473
4	Grass-trees	10	720
5	Hay-windrowed	10	468
6	Soybean-notill	10	962
7	Soybean-mintill	10	2445
8	Soybean-clean	10	583
9	Woods	10	1255
10	Bldg-Grass-Tree-Drives	10	376

Table 3: The ground truth classes and their respective training and testing numbers of the Pavia University datasets.

No.	Name of class	Training	Testing
1	Asphalt	10	6621
2	Meadows	10	18639
3	Gravel	10	2089
4	Trees	10	3054
5	Painted metal sheets	10	1335
6	Bare Soil	10	5019
7	Bitumen	10	1320
8	Self-Blocking Bricks	10	3672
9	Shadows	10	937

Table 4: The ground truth classes and their respective training and testing numbers of the Salinas datasets.

No.	Name of class	Training	Testing
1	Brocoli_green_weeds_1	10	1999
2	Brocoli_green_weeds_2	10	3716
3	Fallow	10	1966
4	Fallow_rough_plow	10	1384
5	Fallow_smooth	10	2668
6	Stubble	10	3949
7	Celery	10	3569
8	Grapes_untrained	10	11261
9	Soil_vineyard_develop	10	6193
10	Corn_senesced_green_weeds	10	3268
11	Lettuce_romaine_4weeks	10	1058
12	Lettuce_romaine_5weeks	10	1917
13	Lettuce_romaine_6weeks	10	906
14	Lettuce_romaine_7weeks	10	1060
15	Vineyard_untrained	10	7258
16	Vineyard_vertical_trellis	10	1797

MNF-ARMG can obtain better results, such as objects 2#, 3#, 6#, and 8# in Table 6.

(3) The results on the Salinas dataset show that all methods show close results in AA, but SS-MNF-ARMG has a competitive advantage in OA and kappa. From Table 7, we can observe that, compared with R-VCANet, SS-MNF-ARMG obtains a 0.1% improvement in OA, a 3.13% improvement in AA, and a 7.31% improvement in kappa. Furthermore, the proposed SS-MNF-ARMG surpasses iCapsNet by 11.01% in OA. In addition, for the classes that other methods do not recognize accurately, SS-MNF-ARMG can obtain better results, such as objects 3#, 7#, 8#, 9#, 15#, and 16# in Table 7.

(4) In general, the decision fusion-based methods outperform the segmentation-based methods and feature fusion-based methods. First and foremost, the LBP features improve the performance of the SS-MNF-ARMG, and the application of MNF reduces the HSI data dimension, controlling the noise in the

Table 5: Classification results of different techniques for the Indian Pines dataset (10 classes).

No.	Pixon-based	PCA-SPCA-2D-SSA	R-VCANet	RN-FSC	iCapsNet	RPNet	MBFSDA	SS-MNF-ARMG
1	41.42	61.55	54.94	72.29	64.71	60.32	70.57	**75.38**
2	33.93	86.49	74.09	78.97	95.56	73.38	81.06	**95.6**
3	71.43	99.32	83.54	90.59	82.09	93.56	81.09	**100.00**
4	23.16	96.05	94.78	89.82	**99.06**	84.2	82.06	93.2
5	87.73	99.37	96.13	99.79	99.59	96.73	99.39	**99.79**
6	65.81	87.97	75.55	86.61	73.35	86.47	77.48	**93.53**
7	76.09	86.28	61.43	89.94	88.01	81.48	66.37	**92.43**
8	22.96	97.01	80.64	**98.25**	74.59	60.1	73.94	97.69
9	95.44	**98.85**	88.15	82.61	98.84	58.73	90.8	97.63
10	91.58	93.52	82.63	94.82	**96.84**	95.79	59.21	94.82
AA	60.96	90.64	79.19	88.37	87.27	79.08	78.2	**93.01**
OA	62.45	87.27	86.28	86.16	85.83	75.94	76.42	**91.05**
Kappa	56.7	85.3	72.05	83.97	83.63	72.32	72.95	**89.61**

Bold values represent the best results among these methods.

Table 6: Classification results of different techniques for the Pavia University dataset (9 classes).

No.	Pixon-based	PCA-SPCA-2D-SSA	R-VCANet	RN-FSC	iCapsNet	RPNet	MBFSDA	SS-MNF-ARMG
1	45.20	**94.83**	79.09	87.28	65.84	83.52	86.68	70.83
2	82.00	81.91	71.76	84.33	84.55	54.19	89.22	**96.78**
3	90.23	85.47	85.45	90.42	29.91	70.70	87.28	**96.24**
4	74.05	87.99	**94.28**	78.09	82.70	91.64	79.34	66.42
5	85.13	97.55	99.98	99.56	**100.00**	**100.00**	99.33	96.51
6	18.55	71.55	58.44	63.25	40.22	65.30	89.52	**100.00**
7	92.48	**98.77**	94.52	52.09	21.40	92.41	88.12	95.49
8	84.17	79.90	73.70	84.81	57.19	74.93	68.88	**95.11**
9	97.57	96.30	**99.82**	95.94	0.00	93.24	82.89	77.51
AA	74.38	88.25	84.12	81.94	66.63	80.66	85.70	**88.32**
OA	69.63	84.50	83.23	81.75	53.54	68.81	86.45	**90.32**
Kappa	57.72	79.86	67.97	75.84	56.07	61.78	82.31	**87.26**

Bold values represent the best results among these methods.

Table 7: Classification results of different techniques for the Salinas dataset (16 classes).

No.	Pixon-based	PCA-SPCA-2D-SSA	R-VCANet	RN-FSC	iCapsNet	RPNet	MBFSDA	SS-MNF-ARMG
1	**99.90**	96.30	96.95	99.26	93.41	98.01	96.95	97.11
2	**99.38**	88.94	99.06	93.21	67.16	99.19	97.65	98.98
3	96.20	92.09	97.31	97.87	86.49	96.91	99.60	**100.00**
4	91.61	90.23	99.46	85.50	90.36	99.21	**100.00**	86.21
5	86.89	88.94	97.35	84.81	90.39	**98.36**	96.30	88.67
6	88.79	96.08	**99.87**	99.35	99.02	99.60	99.14	89.47
7	97.32	91.36	95.26	96.65	95.07	96.42	93.55	**97.65**
8	87.20	96.44	64.01	66.24	82.12	76.01	78.33	**96.66**
9	99.84	85.97	97.79	97.34	98.63	96.58	97.81	**100.00**
10	46.34	86.38	81.38	82.66	93.13	84.66	**94.23**	87.22
11	52.90	94.17	93.17	73.96	60.09	96.25	**97.38**	88.30
12	30.10	80.49	96.48	96.84	94.78	97.87	**99.27**	97.18
13	89.74	96.17	**98.70**	85.39	49.51	93.56	98.69	86.42
14	84.21	**99.30**	90.35	90.39	76.76	97.76	75.70	95.79
15	94.79	86.68	56.92	65.85	49.76	66.26	80.38	**95.36**
16	92.09	98.17	90.79	86.89	96.89	97.57	99.89	**100.00**
AA	83.58	91.43	90.93	87.64	77.26	93.39	94.05	**94.06**
OA	87.15	91.73	93.56	93.12	82.72	88.16	90.96	**93.73**
Kappa	85.63	90.90	85.72	85.44	74.72	86.83	89.95	**93.03**

Bold values represent the best results among these methods.

HSI data. In addition, the randomness in the SS-MNF-ARMG can be regarded as a regularization technique, which alleviates the phenomenon of overfitting. Finally, benefitting from the semi-supervised deep learning framework, the classification accuracy is relatively satisfactory even with only a few labeled samples.

Our experiments were performed using MATLAB 2018b on a computer with an IntelCore(TM) i5-4300M 2.60 GHz CPU, 16 GB memory, and a 64-bit Windows 7 system. For the three real HSI datasets, the duration to execute our algorithm was several minutes to several hours; notably, the duration to execute the other methods that we reviewed in this study was shorter than that of our method.

5. Conclusions and Further Research

In this study, we developed a novel decision fusion method for HSI data classification. In the proposed SS-MNF-ARMG, MNF and multiscale LBP were integrated to extract local SSFs. On the one hand, MNF helped reduce the dimension of the HSI, remove the noise in the HSI, and extract the spectral features from the HSI data. On the other hand, multiscale LBP was applied to the MNF dimension-reduced images to derive spatial features at different scales. These spatial features were further fused with the MNF spectral feature to form the SSFs. Compared with some state-of-the-art spectral-spatial classification methods, our experimental results have demonstrated that SS-MNF-ARMG can achieve higher classification accuracy with limited training samples. This method is effective for distinguishing different land cover types. In addition, a set of optimal parameters for different hyperspectral data can be obtained automatically.

Although the SS-MNF-ARMG algorithm has provided promising results, the classification accuracies of various datasets remain different. Further research could attempt to further improve the generalization ability of our method. Due to human ecological destruction or natural disasters (e.g., earthquakes), the vegetation in some areas has significantly changed. Monitoring vegetation restoration in these areas is of substantial importance. Therefore, we plan to research the application of the HSI classification in vegetation restoration monitoring.

Data Availability

The data used to support the findings of this study are available at GIC (http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes).

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 61703060, 61802036, 61701048, and 61873305, the Sichuan Science and Technology Program under Grant no. 21YYJC0469, the Project funded by China Postdoctoral Science Foundation under Grant no. 2020M683274, the Fundamental Research Funds of the Central Universities, Southwest Minzu University (2019NQN07), the Opening Fund of Geomathematics Key Laboratory of Sichuan Province (scsxdz2020zd01 and csxdz201710), and the Key Laboratory of Pattern Recognition and Intelligent Information Processing of Sichuan (MSSB-2018-0 and MSSB-2020-9).

References

[1] T. Matsuki, N. Yokoya, and A. Iwasaki, "Hyperspectral tree species classification of Japanese complex mixed forest with the aid of LiDAR data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 5, pp. 2177–2187, 2015.

[2] P. Hardin and A. Hardin, "Hyperspectral remote sensing of urban areas," *Geography Compass*, vol. 7, no. 1, pp. 7–21, 2013.

[3] Y. Xie, F. Miao, K. Zhou et al., "HsgNet: a road extraction network based on global perception of high-order spatial information," *ISPRS International Journal of Geo-Information*, vol. 8, no. 12, p. 571, 2019.

[4] Z. Pan, J. Liu, L. Ma et al., "Research on hyperspectral identification of altered minerals in Yemaquan West Gold Field, Xinjiang," *Sustainability*, vol. 11, no. 2, p. 428, 2019.

[5] X. Kang, X. Xiang, S. Li et al., "PCA-based edge-preserving features for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7140–7151, 2017.

[6] M. B. Stuart, A. J. S. McGonigle, and J. R. Willmott, "Hyperspectral imaging in environmental monitoring: a review of recent developments and technological advances in compact field deployable systems," *Sensors*, vol. 19, no. 14, p. 3071, 2019.

[7] X. Briottet, Y. Boucher, A. Dimmeler et al., "Military applications of hyperspectral imagery, Targets and backgrounds XII: characterization and representation," *International Society for Optics and Photonics*, vol. 6239, p. 62390B, 2006.

[8] B. Hörig, F. Kühn, F. Oschütz et al., "HyMap hyperspectral remote sensing to detect hydrocarbons," *International Journal of Remote Sensing*, vol. 22, no. 8, pp. 1413–1422, 2001.

[9] C. Butz, M. Grosjean, D. Fischer et al., "Hyperspectral imaging spectroscopy: a promising method for the biogeochemical analysis of lake sediments," *Journal of Applied Remote Sensing*, vol. 9, no. 1, Article ID 096031, 2015.

[10] K. Jacq, Y. Perrette, B. Fanget et al., "Hyperspectral imaging for lake sediment cores analysis," in *Proceedings of the IPAIAL Stockholm 2018 Conference*, pp. 735–743, Stockholm, Sweden, June 2018.

[11] E. C. Koeniguer, F. Janez, and J. M. Nicolas, "Change detection in SAR time-series based on the coefficient of variation," 2019, https://arxiv.org/abs/1904.11335.

[12] M. Zhang, J. Zhou, K. A. Sudduth et al., "Estimation of maize yield and effects of variable-rate nitrogen application using UAV-based RGB imagery," *Biosystems Engineering*, vol. 189, pp. 24–35, 2020.

[13] M. P. Uddin, M. A. Mamun, M. I. Afjal et al., "Information-theoretic feature selection with segmentation-based folded principal component analysis (PCA) for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 42, no. 1, pp. 286–321, 2021.

[14] H. Fu, G. Sun, J. Ren et al., "Fusion of PCA and segmented-PCA domain multiscale 2-D-SSA for effective spectral-spatial feature extraction and data classification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[15] C. Ruichun, W. Lu, and W. Maozhi, "Application of FastICA in mineral information extraction from hyperspectral remote sensing data," *Remote Sensing for Land & Resources*, vol. 25, no. 4, pp. 129–132, 2013.

[16] A. A. Green, M. Berman, P. Switzer et al., "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 26, no. 1, pp. 65–74, 1988.

[17] W. Li, S. Prasad, J. E. Fowler et al., "Locality-preserving dimensionality reduction and classification for hyperspectral image analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1185–1198, 2011.

[18] W. Zhao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544–4554, 2016.

[19] A. Zehtabian and H. Ghassemian, "Automatic object-based hyperspectral image classification using complex diffusions and a new distance metric," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 7, pp. 4106–4114, 2016.

[20] B. Pan, Z. Shi, X. Xu, and R-VCANet, "A new deep-learning-based hyperspectral image classification method," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 5, pp. 1975–1986, 2017.

[21] M. Imani and H. Ghassemian, "Discriminant analysis in morphological feature space for high-dimensional image spatial–spectral classification," *Journal of Applied Remote Sensing*, vol. 12, no. 1, Article ID 016024, 2018.

[22] A. Sellami, A. B. Abbes, V. Barra et al., "Fused 3-D spectral-spatial deep neural networks and spectral clustering for hyperspectral image classification," *Pattern Recognition Letters*, vol. 138, pp. 594–600, 2020.

[23] M. Imani and H. Ghassemian, "An overview on spectral and spatial information fusion for hyperspectral image classification: current trends and challenges," *Information Fusion*, vol. 59, pp. 59–83, 2020.

[24] X. Huang and L. Zhang, "An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 257–272, 2012.

[25] K. Y. Peerbhay, O. Mutanga, and R. Ismail, "Random forests unsupervised classification: the detection and mapping of solanum mauritianum infestations in plantation forestry using hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 3107–3122, 2015.

[26] K. Shi, J. Wang, Y. Tang et al., "Reliable asynchronous sampled-data filtering of T–S fuzzy uncertain delayed neural networks with stochastic switched topologies," *Fuzzy Sets and Systems*, vol. 381, pp. 1–25, 2020.

[27] H. Han, G. Wang, M. Wang et al., "Hyperspectral Unmixing via nonconvex sparse and low-rank constraint," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5704–5718, 2020.

[28] W. Dong, F. Fu, G. Shi et al., "Hyperspectral image super-resolution via non-negative structured sparse representation," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2337–2352, 2016.

[29] Y. Xu, B. Du, F. Zhang et al., "Hyperspectral image classification via a random patches network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 142, pp. 344–357, 2018.

[30] J. Yin, S. Li, H. Zhu et al., "Hyperspectral image classification using CapsNet with well-initialized shallow layers," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 7, pp. 1095–1099, 2019.

[31] K. Gao, B. Liu, X. Yu et al., "Deep relation network for hyperspectral image few-shot classification," *Remote Sensing*, vol. 12, no. 6, p. 923, 2020.

[32] P. Ghamisi, R. Souza, J. A. Benediktsson et al., "Hyperspectral data classification using extended extinction profiles," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 11, pp. 1641–1645, 2016.

[33] L. Fang, N. He, S. Li et al., "A new spatial–spectral feature extraction method for hyperspectral images using local covariance matrix representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3534–3546, 2018.

[34] C. Ge, Q. Du, W. Li et al., "Hyperspectral and LiDAR data classification using kernel collaborative representation based residual fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 6, pp. 1963–1973, 2019.

[35] W. N. Khotimah, M. Bennamoun, F. Boussaid et al., "A high-performance spectral-spatial residual network for hyperspectral image classification with small training data," *Remote Sensing*, vol. 12, no. 19, p. 3137, 2020.

[36] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.

[37] R. Girshick, J. Donahue, T. Darrell et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[39] C. Qi, Z. Zhou, Y. Sun et al., "Feature selection and multiple kernel boosting framework based on PSO with mutation mechanism for hyperspectral classification," *Neurocomputing*, vol. 220, pp. 181–190, 2017.

[40] X. Ceamanos, B. Waske, J. A. Benediktsson et al., "A classifier ensemble based on fusion of support vector machines for classifying hyperspectral data," *International Journal of Image and Data Fusion*, vol. 1, no. 4, pp. 293–307, 2010.

[41] Y. Gu and H. Liu, "Sample-screening MKL method via boosting strategy for hyperspectral image classification," *Neurocomputing*, vol. 173, pp. 1630–1639, 2016.

[42] J. Jiang, J. Ma, C. Chen et al., "SuperPCA: a superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4581–4593, 2018.

[43] Q. Zhang, J. Sun, G. Zhong et al., "Random multi-graphs: a semi-supervised learning framework for classification of high dimensional data," *Image and Vision Computing*, vol. 60, pp. 30–37, 2017.

[44] F. Gao, Q. Wang, J. Dong et al., "Spectral and spatial classification of hyperspectral images based on random multi-graphs," *Remote Sensing*, vol. 10, no. 8, p. 1271, 2018.

[45] H. Chen, F. Miao, and X. Shen, "Hyperspectral remote sensing image classification with CNN based on quantum genetic-

optimized sparse representation," *IEEE Access*, vol. 8, pp. 99900–99909, 2020.

[46] F. Bianconi, R. Bello-Cerezo, and P. Napoletano, "Improved opponent color local binary patterns: an effective local image descriptor for color texture classification," *Journal of Electronic Imaging*, vol. 27, no. 1, Article ID 011002, 2017.

[47] B. Tu, W. Kuang, G. Zhao et al., "Hyperspectral image classification by combining local binary pattern and joint sparse representation," *International Journal of Remote Sensing*, vol. 40, no. 24, pp. 9484–9500, 2019.

[48] F. R. K. Chung, "Lectures on spectral graph theory," *CBMS Lectures, Fresno*, vol. 6, no. 92, pp. 17–21, 1996.

[49] W. Liu, J. Wang, S. Kumar et al., "Hashing with graphs," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pp. 1–8, Bellevue, WA, USA, June-July 2011.

[50] S. Kim and S. Choi, "Multi-view anchor graph hashing," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3123–3127, IEEE, Vancouver, Canada, May 2013.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 9948855, 11 pages
https://doi.org/10.1155/2021/9948855

Research Article

Modeling and Analysis of Network Control System Based on Hierarchical Coloured Petri Net and Markov Chain

Jingdong Li,[1,2] **Zhangang Wang**,[1,2] **Liankun Sun**,[1,2] **and Wanru Wang**[1,2]

[1]*School of Computer Science and Technology, Tiangong University, Tianjin 300387, China*
[2]*Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems, Tiangong University, Tianjin 300387, China*

Correspondence should be addressed to Zhangang Wang; wangzhangang@tiangong.edu.cn

Received 8 March 2021; Revised 24 April 2021; Accepted 28 May 2021; Published 9 June 2021

Academic Editor: Zi-Peng Wang

This paper investigates a modified modeling of networked control systems (NCSs) with programmable logic controller (PLC). First, the controller-to-actuator and sensor-to-controller network-induced delays are investigated by a modeling tactics based on hierarchical coloured petri net (HCPN) in a structure-conserving way. Comparing with the recent result, the signal transmission delay is set in a random interval instead of a fixed mode; moreover, the data packet drop out and disorder are also taken into consideration. Second, delays captured form CPN tools are analyzed with a strategy based on Baum–Welch algorithm and statistics science. Besides, time delays are modeled as a Markov chain and the transition probabilities is calculated using the consequent from the previous operation. Finally, a comparison verification illustrates the equivalence property between proposed models.

1. Introduction

As a kind of control system [1–6], NCSs have motivated a lot of research studies in the control field during recent years [7–10] by many advantages it shares; for example, reduced weight, high flexibility, simple installation and maintenance, and low cost. Because of those practical characteristics, more and more efforts have been devoted to these systems [11–13]. Admired for past achievements, NCSs have been found in widespread applications such as feedback control systems [14, 15], stabilization of linear systems [16], control of nonlinear systems [17], and adaptive tracking control of nonlinear multiagent systems [18–20]. Compared with traditional point-to-point hardwiring system, typical NCSs is a young generation of control architectures which has a feedback control structure consisting with the controller, sensor, and actuator through network communication. The insertion of this performance condition, with a finite bandwidth, brings serial challenging and undesirable issues on account of data packet loss, scheduling, and latency in the process of communication signal's transmission to remote analog input/output unit. Affected by these cases, the communication channel, which takes the place of traditional signal transmission technologies, brings hard-to-solve problems in the stability analysis and controller synthesis because of transmission time delays. As a result, this issue received increasing interests in this field [21–23]. Traditional time-delay analysis usually assumes that the time delay is constant, time-varying or obeys some random distribution, and rarely analyzes it from the perspective of system operation mechanism. Modeling from the perspective of operation mechanism can effectively show various state changes that may occur in the system and the relationship between changes. This prompts us to adopt HCPN to carry out structural reservation modeling and then analyze the time-delay characteristics.

In 1992, Petri nets were introduced by Petri [24] firstly as a net-theoretic approach to implement a particular purpose. The relationships between departments could be represented by a net, and it is a good approximation to imitate the appearance and character of asynchronous and concurrent operation in discrete event systems. Petri nets is a kind of mathematical structures which is a bipartite-directed graph consisting of two kinds of compositions; one is drawn as a

circle which is called Place, and another drown as a rectangles named Transition. These two nodes combine with each other with arcs which are drawn as arrows. Coloured Petri net (CPN) [25–28], in which groups of objects is thus distinguished by colour, inherits all the advantages of classical Petri net. The wide practical application distributes on the direction of distributed control systems [29] and domains environments [30]. Recently, a novel of theoretical results has been done on the application of Petri nets for NCSs [31, 32]. Regarding the execution of the NCSs practical operational fundamental principle, the CPN has been chosen to build a formalism model [31]. With the rapid development of scientific technologies, the process in large-scale systems is more and more complicated, which makes it hard to solve problems in modeling. Therefore, to solve the problem of state explosion during formal verification, HCPN is invented [33–35] which is a new type of Petri net for creating large-scale and complex systems. Its main purpose is to summarize the system model with simple network model and to expand and fill it with substitution change. This method is not only beneficial to the excessive number of reservoirs, transitions, and arcs in the model but also beneficial to avoid the explosion of system state space and simplify the analysis of the model. In [31], a mapping from the established hypothetical delays in Ethernet of NCSs to entities of a CPN model was defined, and the simulation analysis of network induced delays was tested and verified in a simple way. Following a similar way, in [32], a two-step approach was included in the estimation of delays in the modeling of NCSs. However, the Ethernet model in the proposed model was studied in a hypothesis, in which time delays were not interrelating with network bandwidth-limited bit limit and packet dropouts. Further investigation and analysis is in [36, 37] and the references therein.

The primary contributions of this paper are as follows:

(i) First, considering the potential cause for the deteriorating performance or instability on NCSs, this paper sets network band delay in an interval instead of the definite transmission delay in [32] to make the model more realistic. In this method, it is supported by the HCPN model to simulate some challenging issues on account of network information propagation.

(ii) Second, the CPN model in [31, 32] is not considered data packet latency and dropout in the Ethernet CPN model of NCSs. Focusing on those challenges in network transmission, this paper has some alterations on the base of the model in [32]; close to the reality, data packets' dropout and packet out-of-order are taken into consideration in the proposed model of data transmission on the network.

(iii) Third, the exploration and research of random time delays existing in the sensor-to-controller and controller-to-actuator in discrete time networked control systems are acquiescence with random Markovian delays in recent literature [38–40]. However, it is not clear if this theoretical acquiescence can be put into practice. This paper makes a certification of equivalence between time delays and Markov model [41].

This paper is organized as follows. Section 2 is the conceptual framework of Petri net. Section 3 is the HCPN model of NCSs-PLC. Section 4 analyzes the time delay extracted from the HCPN model. Section 5 is the equivalence analysis of the time delay of both the Markov model and the HCPN model.

2. Basic Conception

Definition 1. A ordinary non-HCPN can be defined as a nine elements' tuple [42],

$$\text{Traditional CPN} = (\Sigma, P, T, A, N, C, G, E, I), \tag{1}$$

satisfying the requirements below:

(i) Σ is called colour set which is a finite set describing nonempty types.

(ii) P is Place representing a ellipse which interprets a passive component with discrete status.

(iii) T is Transition establishing with a rectangle which explains an active component; tokens can consume, produce, and change the carrying information in Transitions.

(iv) A is arcs which connects Places and Transitions in the model. It can be represented by arrows, and it is a finite set which meets the expectations with $\forall P_i \in P; T_i \in T$: $P_i \cap T_i = P_i \cap A_i = T_i \cap A_i = \varnothing$.

(v) N is defined as a map into arcs. It is defined from A into $P_i \times T_i \cup T_i \times P_i$ which has two elements, the first element means arcs' source and the second element means arcs' destination.

(vi) C is defined a map in places. It is defined from P_i into Σ which means that every token on every P_i has a corresponding colour set type.

(vii) G is the guard function which is defined a map in transitions. $G(T_i)$ is the type appertain to Σ, and the binding must perform every Boolean expressions. It is can be shown as $\forall T_i \in T$ [Type (Var (Guard (T)))$\subseteq\Sigma\wedge$Type (Guard (T_i)) = Boolean].

(viii) E is called arc expression which maps every element in A to an expression, and the type of it can be written as $C(P_i)_{\mathrm{MS}}$; it can be shown as

$$\forall a = A\left[\mathrm{Type}(E_i) = C\left(\widehat{P}_i\right)_{\mathrm{MS}} \wedge \mathrm{Type}\left(\mathrm{Var}(E_i)\right) \subseteq \Sigma\right], \tag{2}$$

where $\widehat{P}_i$ is defined as the place of N_i.

(ix) I illustrates a map between P_i and the type $\widehat{C}_i$ It is the initialization function which can be shown as

$$\forall P_i \in P\colon \left[\mathrm{Type}(I_i) = \widehat{C}_i\right]. \tag{3}$$

Definition 2. A HCPN can be defined as a nine elements-tuple satisfying the requirements below [42]:

$$\mathrm{HCPN} = (S_e, S_eN, S_eA, \mathrm{PL_{PN}}, \mathrm{PO}_T, \mathrm{PO}_A, \mathrm{PL_{FS}}, \mathrm{FT}, \mathrm{PP}). \tag{4}$$

(i) S_e is a congregations for nonhierarchical pages in the model, and each page does not have collaborative net elements. It can be shown as

$$\left[\Sigma_{S_i}, P_{S_i}, T_{S_i}, A_{S_i}, N_{S_i}, C_{S_i}, G_{S_i}, E_{S_i}, I_{S_i}\right], \quad \forall S_1, S_2 \in S_e: \left[S_1 \neq S_2 \Rightarrow \left(P_{S_1} \cup T_{S_1} \cup A_{S_1}\right) \cap \left(P_{S_2} \cup T_{S_2} \cup A_{S_2}\right) = \varnothing\right]. \tag{5}$$

(ii) $S_eN \in T$ is substitution nodes.

(iii) S_eA is called page assignment, and every page must not be the subpage of itself; it is satisfied by the following conditions:

$$\left\{\widehat{s}_0, \widehat{s}_1, \ldots, \widehat{s}_i \in \widehat{S}_i | n \in N^+ \wedge \widehat{s}_0 = \widehat{s}_n \wedge \forall k \in 1, \ldots, n\colon s_k \in S_eA\left(\mathrm{SN}_{S_{i(K-1)}}\right)\right\} = \varnothing. \tag{6}$$

(iv) $\mathrm{PL_{PN}} \subseteq P$ is defined as a set of place nodes or transition nodes.

(v) POT is a function of port type, and it has four types, in type, out type, in/out type, and general.

(vi) PO_A is port assignment. It is binary relations such that

$$\begin{aligned} &\forall t \in S_eN\colon \mathrm{PO}_A(t) \subseteq X(t) \times \mathrm{PL}_{\mathrm{PN}-S_eA_{(T)}}, \\ &\forall t \in S_eN\colon \forall\left(p_1, p_2\right) \in \mathrm{PO}_A\colon \mathrm{PO}_T(p_2) \neq \text{general} \Rightarrow \mathrm{ST}(p_1, t) = \mathrm{PT}(p_2), \\ &\forall t \in S_eN\colon \forall\left(p_1, p_2\right) \in \mathrm{PO}_A\colon [C(p_1) = C(p_2) \wedge I(p_1) = I(p_2)]. \end{aligned} \tag{7}$$

(vii) $\mathrm{PL_{FS}} \in P_s$ is a finite set of fusion sets such that

$$\begin{aligned} &\forall fs \in \mathrm{PL_{FS}}\colon \\ &\forall p_1, p_2 \in fs\colon [C(p_1) = C(p_2) \wedge I(p_1) = I(p_2)]. \end{aligned} \tag{8}$$

(viii) FT is a function of the fusion type. It is defined from fusion into {global, page, instance}, such that

$$\begin{aligned} &\mathrm{FT}\colon \mathrm{FS} \longrightarrow \{\text{global, page, instance}\}, \\ &\forall fs \in \mathrm{FS}\colon [\mathrm{FT}(fs) \neq \text{global} \Rightarrow \exists s \in S\colon fs \subseteq Ps]. \end{aligned} \tag{9}$$

(ix) $\mathrm{PP} \in S_{\mathrm{MS}}$ is a multiset of the prime page.

3. The CPN Model for NCSs

3.1. The Top Model. The integrated model can be divided into two levels, one is top level which simulates the relationship between the controller, the sensor, and the actuator with a cursory method, as shown in Figure 1. The other level is the detailed function description of the controller, the sensor, and the actuator. In the proposed top-level model, the sensor and the actuator connect with PLC Remote IO. PLC-CPU sends packets from the controller to PLC Remote IO through Ethernet; then, PLC Remote IO sends acknowledgments from the sensor and the actuator back to PLC-CPU. There are three subpages and one top page in the whole model. After some introductions of time delay in PLC-CPU, those pages are explained one by one in detail.

3.2. The Analysis of the Time Delays in PLC-NCSs. Figure 2 is a representative of traditional NCSs with the PLC controller. The sensor and actuator are connected to digital/analog I/O section through Ethernet. Before entering or leaving network, data packets will enter into a transmission queue to wait for the scan cycle of I/O.

Figure 3 gives detailed instructions of time delays in NCSs with the PLC controller, in which different samples were shown as different length arrows. In the upper part of the whole figure, there are some samples which are received in the sensor with periodic sampling. In the middle part, there are three kinds of axis showing the PLC controller. Before data packets arrive at execute program, it has to wait for the scan cycle so that it can be read into PLC. Both of the reading scan cycle and writing scan cycle in PLC communication are Tc. Another primary scan cycle is Tp which means that only at Tp scan cycles data packets can be captured into PLC-CPU to execute the programme. After a few execution time (τ_c), the communication module sends result data to the actuator node. During the process of the data packets' transmission, two network-induced delays are produced. One is called backward time delay and another is called forward time delay. The execution time delay in the controller is included in Tsc backward time delay which is the time costing in the transmission from the controller to the sensor, and Tca is forward time delay which is from CPU which received the sample to the actuator which received the same sample.

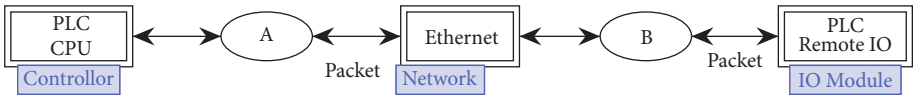


Figure 1: The top level in the CPN model for NCSs.

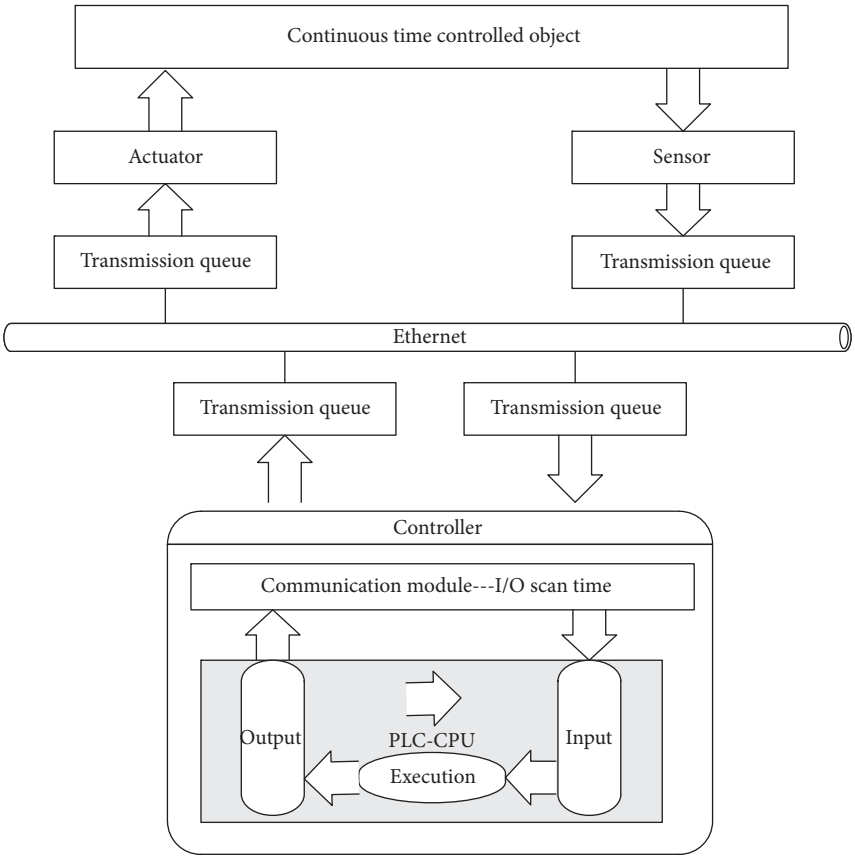


Figure 2: Network control system with PLC being the CPU controller.

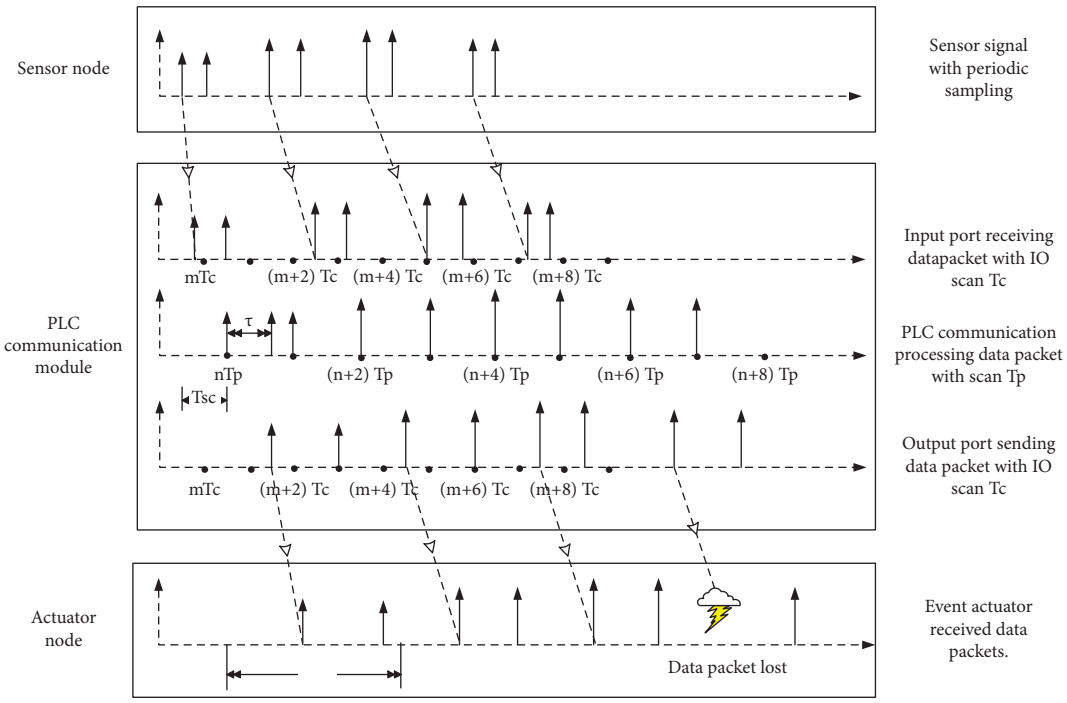


Figure 3: Process of time delay in PLC-CPN.

3.3. The Controller Model. The module of controller is divided into two parts. One is the communication module part, which is used to sending and receiving data with scan cycles, such as left part in Figure 4. The another part is the PLC-CPU model, which simulates functions of CPU reading(input), CPU executing, and CPU writing(output), such as right part in Figure 4. The Send_R transition and the Rec_A transition complete data packets' sending and receiving, respectively. PLC_R transition, PLC_E transition, and PLC_W transition complete simulation of reading, executing, and writing in the PLC-CPU. The definition of token elements in the model is shown in Table 1.

3.4. The Ethernet Model. Figure 5 is the Ethernet model which is the primary part for whole NCSs' system. There are two modified points contrasting with the model which has been designed by Ghanaim et al. [32]. When the data packets transmit in Ethernet, transmission will be influenced by a lot of uncertainties. In consideration of these cases, the transmission time is set into a random time with an interval time varying rather than a fix data. In the proposed model, time delays of this portion are discrete uniform distributions between 1000 and 2000 when packet data passed from the sensor to the controller or from the controller to the actuator. In addition, in the actual network transmission, the data packets' transmission is unstable because of some uncertain factors, such as packets lost and packets out of order. In this model, packets will be lost in ten percent probability, and this function is realized by fun Ok $(s\colon \text{Ten0}, r\colon \text{Ten1}) = (r <= s)$. Data packets will be retransmitted based on Transmission Control Protocol/Internet Protocol if those transmission mistakes occurred. The mathematic presentation of this model is a tuple:

$$\text{HCPN} = (S.SN.SA.PN, \text{PT}, \text{PA}, \text{FS}, \text{FT}, \text{PP}). \tag{10}$$

By the definition of HCPN, those elements in this tuple can be used to describe the Ethernet model, as shown in Figure 6.

Σ is the Token colour set in Figure 5. P is finite set of places and T is finite set of transitions. A is arc, and C to Store1 means that this arc is from place C to transition Store1. N is the node function. G is guard, for example, $(\#A, p1) = N, T = \text{Net1}, \text{Net2}$ means that when transition is Net1 or Net2, the guard of transition is $(\#A, p1) = N$; otherwise, transition guards are true. C is the colour definition of Place. Place A, B, C, and D are mean the Packet colour. $I(P)$ is the initialization set if the place is $SP1$ or $SP2$, and the initial token is integer nine. $E(a)$ means the arc expression in the Ethernet model, $p1, a \in [A \text{ to Net1}, C \text{ to Store1} \cdot \text{Store1 to } A]$ means when the arc is from A to Net1, C to Store 1, and Store1 to A, and the arc expression on those arc is $p1$.

4. Time-Delay Analysis

4.1. Transmission Delay in Ethernet. Network transmission is an important element in the analysis of Tsc and Tca. Besides, it also can help to analyze the system state and performance. Data packets may be lost at a probability such as ten percent in the model of network transmission. With reference to Transmission Control Protocol/Internet Protocol (TCP/IP), data packets would be retransmitted if it is lost in the network transmission. Figure 7 is network transmission latency which is collected in the proposed HCPN model running 5050 steps. From Figure 7, it can be clearly see that, in time delay Tsc and Tca, network transmission time delays are almost greater than 1000 and less than 2000; however, there are still ten percent points out of scope, Figure 7 shows several peak points, for example A, B, and C, which mean that, in this time, data packets are dropped out in network transmission. It will be retransmitted according to TCP/IP so that the value of it becomes very bigger than others. The network transmission latency was the transmission time and the retransmission time and other time spent on extracting data packets.

4.2. The Time Delay Tca and Tsc. There are three scan cycles which play important roles in the calculation of backward time delay and forward time delay. The first one is periodic scan Tp for CPU program, the second one is periodic IO scan Tc for the communication module, and the last one is the periodic sampling T_h for the sensor. There exists another important data to calculate: network transmission delay. Table 2 is the parameters defined in the NCSs model which can be used to simulate and analyze the efficiency of the difference system with the HCPN model.

In addition, in the communication module, IO periodic scan cycle Tc is a summation of reading time, executing time, and writing time. The periodic scan Tp for CPU program is set to 17,000 and the periodic sampling Th for the sensor is 1000. When network transmission latency was obtained, such as in Figure 7, Tca and Tsc can be calculated under the rule of scan cycles, as shown in Figure 8. Before the calculation of Tca and Tsc, it is necessary to obtain the network latency, which are produced in network transmission and waiting time for the scan cycle. Contrasting with delays obtained from the HCPN model in [32], it can be clearly see that because of the indeterminacy transmission phenomenon and the random transmission time delay the periodicity of Tca and Tsc is broken.

5. Verification and Analysis

5.1. Markov Modeling. A Hidden Markov Model (HMM) is a statistical Markov lain with hidden states which cannot be directly visible, but the sequence of observations can give some information of HMM. The sequence of observations alphabet is set to O_{seq}, and a hidden states' alphabet sequence S_{seq} can be cached from O_{seq}. O_{seq} and S_{seq} are shown below:

$$\begin{aligned} O_{\text{seq}} &= \{o_1, o_2, \ldots, o_i, \ldots\}, \\ S_{\text{seq}} &= \{s_1, s_2, \ldots, s_i, \ldots\}, \end{aligned} \tag{11}$$

where $O_i \in O = \{o_1, o_2, \ldots, o_N\}$ is the observation state at time k and $S_i \in S = \{s_1, s_2, \ldots, s_N\}$ is the corresponding state at observation O_k.

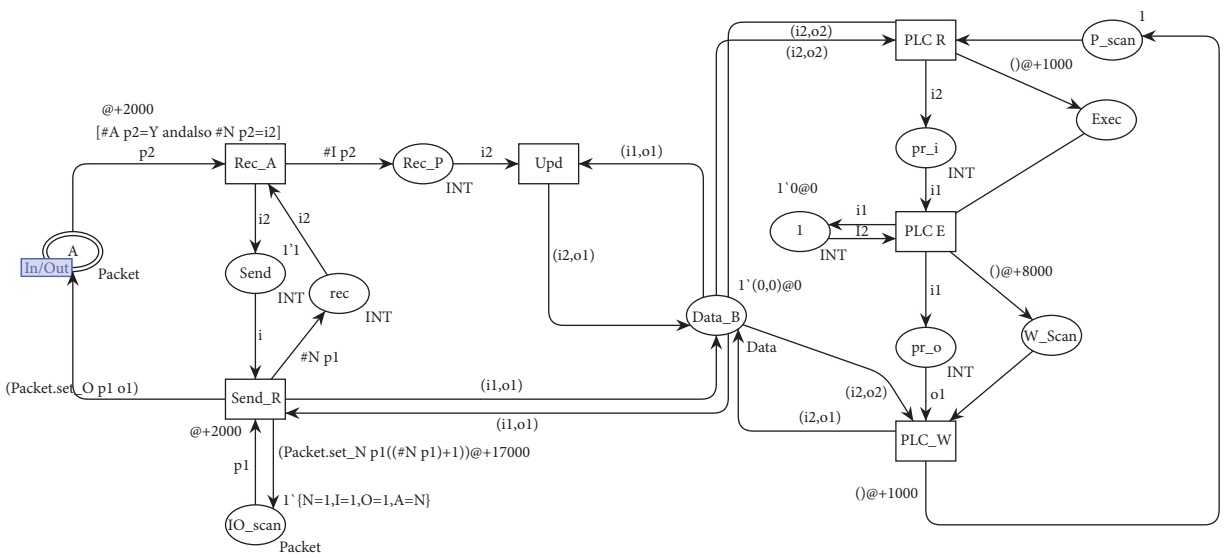


Figure 4: The controller module in the HCPN model for NCSs-PLC.

Table 1: Token definition of the HCPN model for NCSs.

```
colset UNIT = unit timed;
colset BOOL = bool;
colset INT = int timed;
colset TIME = time;
colset STRING = string;
colset N = INT;
colset A = with Y|N;
colset I, O = int;
colset Ten0 = int with 0, . . . , 10;
colset Ten1 = int with 1, . . . , 10;
fun Ok (s: Ten0, r: Ten1) = (r < = s);
colset Packet = record N: N ∗ I: I ∗ O: O ∗ A: A timed;
colset Data = product INT ∗ INT timed;;
var i, i1, i2, o1, o2: INT;
var p1, p2: Packet;;
```

The formal definition of a hidden Markov model (HMM) is as follows:

$$\lambda = \{A, B, \pi_0\}, \tag{12}$$

where A is the state transition probability matrix form one state to another state, which can be written as

$$\begin{aligned} A &= \{a_{ij}\}, \\ \{a_{ij}\} &= P\{s_{k+1} = s_j | s_k = s_i\}, \end{aligned} \tag{13}$$

B is the observation probability matrix which can be written as

$$\begin{aligned} B &= \{b_{ij}\}, \\ \{b_{ij}\} &= P\{o_{k+1} = o_j | s_k = s_i\}, \end{aligned} \tag{14}$$

and π_0 is the initial state probability which can be written as

$$\begin{aligned} \pi_0 &= \{\pi_i\}, \\ \{\pi_i\} &= P\{s_1 = s_j\}. \end{aligned} \tag{15}$$

5.2. Markov Model Results. By the definition of HMM and the Baum Welth algorithm, the algorithm steps are as follows.

Step 1 : for a given observation sequence and HMM model, the probability variables of the hidden state S_i are

$$\begin{aligned} \delta_t(i) &= \frac{\alpha_t(i)\beta(i)}{P(O|\lambda)}, \\ P(O|\lambda) &= \sum_{i=1}^{N} \alpha_t(i)\beta(i), \end{aligned} \tag{16}$$

where $\alpha_t(i)$ is called forward probability, where

$$\alpha_t(i) = P(o_1 \ o_2 \ o_3 \ \ldots \ o_{t-1} \ o_t \ , q_t = S_i | \lambda), \tag{17}$$

and $\beta_t(i)$ is called backward probability, where

$$\beta_t(i) = P(o_{t+1} \ o_{t+2} \ o_{t+3} \ \ldots \ o_{T-1} \ o_T \ , q_t = S_i | \lambda), \tag{18}$$

Step 2 : define the probability between the hidden state S_i and the hidden state S_j as follows:

$$\begin{aligned} \phi_t(i, j) &= P(q_t = S_i, q_{t+1} = S_j | O, \lambda), \\ \phi_t(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}, \\ P(O|\lambda) &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \end{aligned} \tag{19}$$

Step 3 : the formula between $\delta_t(i)$ and $\phi_t(i, j)$ is

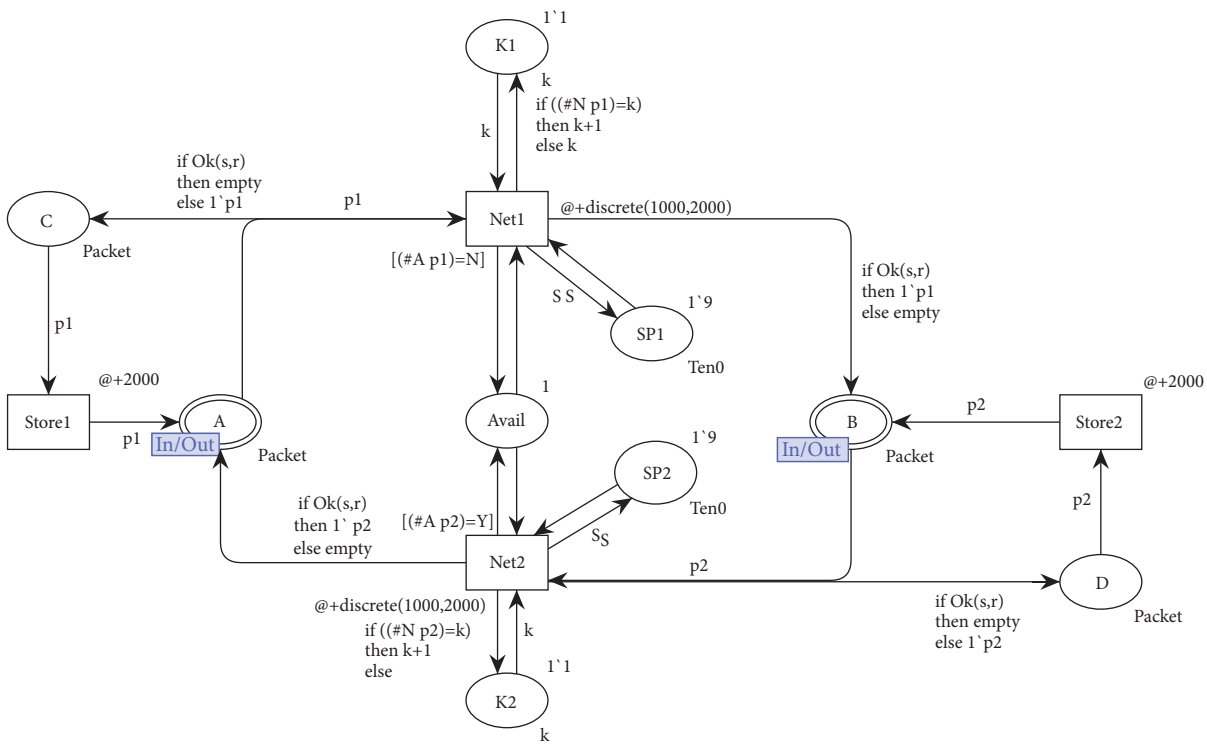


Figure 5: The Ethernet model for NCSs.

$$\delta_t(i) = \sum_{j=1}^{N} \phi_t(i, j). \tag{20}$$

Step 4 : find the corresponding expected values of the above two variables $\sum_{t=1}^{T-1} \delta_t(i)$ and $\sum_{t=1}^{T-1} \phi_t(i, j)$.

Step 5 : the new HMM parameters are estimated by using the two variables defined above and their expected values:

$$\begin{aligned} \hat{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \phi_t(i, j)}{\sum_{t=1}^{T-1} \delta_t(i)}, \\ \hat{b}_j(k) &= \frac{\sum_{t=1}^{T-1} \phi_t(i, j)}{\sum_{t=1}^{T} \delta_t(i)}. \end{aligned} \tag{21}$$

The Markov model can be built based on time delay Tsc and Tcs obtained from $HCPN$. In order to make analyzing more clear, data will be managed with the rounding method. From Figure 8, it two time delays can be obtained as follows:

$$\begin{aligned} Tsc &= [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17], \\ Tca &= [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29]. \end{aligned} \tag{22}$$

Tsc can be divided into two categories: one category is the low delay Tsc_{low} which includes the value $Tsc_{\text{low}} = [3, 4, 5, 6, 7, 8, 9]$. The other category is high delay Tsc_{high} which includes the value $Tsc_{\text{high}} = [10, 11, 12, 13, 14, 15, 16, 17]$.

The time delay Tsc can be written as

$$Tsc = \begin{cases} Tsc_{\text{low}}, & s_k = s_1, \\ Tsc_{\text{high}}, & s_k = s_2. \end{cases} \tag{23}$$

The same procedure can be easily adapted to Tca which can be divided into two categories: one category is the low delay Tca_{low} which includes the value $Tca_{\text{low}} = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]$ and the other category is high delay Tca_{high}, and $Tca_{\text{high}} = [20, 21, 22, 23, 24, 25, 26, 27, 29]$. The time delay Tca can be written as

$$Tca = \begin{cases} Tca_{\text{low}}, & s_k = s_1, \\ Tca_{\text{high}}, & s_k = s_2. \end{cases} \tag{24}$$

Figures 9 and 10 show the probability distribute of time delay Tsc and Tca which are the experimental data in HCPN, respectively.

The parameters in the HMM model can be calculated with Baum–Welch algorithm and maximum likelihood estimate, and the system $\lambda_{sc} = \{A_{sc}, B_{sc}, \pi_{0_{SC}}\}$ with 2-state and 15-observations, and the matrix A_{sc}, B_{sc} can be calculated by Matlab:

$$\Sigma = [UNIT, BOOL, INT, TIME, STRING, Packet, Data, Ten0, Ten1, N, A, I, O]$$

$$P = [C, A, K1, K2, SP1, SP2, avail, B, D] \quad T = [Store1, Store2, Net1, Net2]$$

$$A = \begin{cases} CtoStore1 & Store1toA & AtoNet1 & Net1toC \\ BtoNet2 & Net2toD & DtoStore2 & Store2toB \\ Net1toB & Net2toA & K1toNet1toK1 & Net1toSP1toNet1 \\ Net2toSP2toNet2 & K2toNet2toK1 & Net1toAvailtoNet1 & Net2toAvailtoNet2 \end{cases}$$

$$N = (Source\ dest) \qquad G = \begin{cases} (\#A, p1) = \mathrm{N} & T = Net1, Net2 \\ true & otherwise \end{cases}$$

$$C = \begin{cases} Packet & A, B, C, D \\ INT & K1, K2 \\ Ten0 & SP1, SP2 \\ UNIT & avail \end{cases} \qquad I(p) = \begin{cases} 1 & p = K1, K2 \\ 9 & P = SP1, SP2 \\ () & P = avail \\ \emptyset & other \end{cases}$$

$$E(a) = \begin{cases} p0 & a \in [AtoNet1, CtoStore1.Store1toA] \\ p2 & a \in [BtoNet2, DtoStore2.Store2toB] \\ k & a \in [K1toNet1, K2toNet2] \\ s & a \in [Net1toSP1, SP1tpNet1, Net2toSP2, SP2tpNet2] \\ \text{if } OK(s, r) \text{ then else } empty \text{ else } 1`P1 & a = Net1toC \\ \text{if } OK(s, r) \text{ then else } empty \text{ else } 1`P2 & a = Net2toD \\ \text{if } OK(s, r) \text{ then } 1`P1 \text{ else } empty & a = Net1toB \\ \text{if } OK(s, r) \text{ then } 1`P2 \text{ else } empty & a = Net2toA \\ \text{if } ((\#NP1) = K) \text{ then } k + 1 \text{ else } k & a = Net1toK1 \\ \text{if } ((\#NP2) = K) \text{ then } k + 1 \text{ else } k & a = Net2toK2 \\ () & other \end{cases}$$

Figure 6: Mathematical description for HCPN.

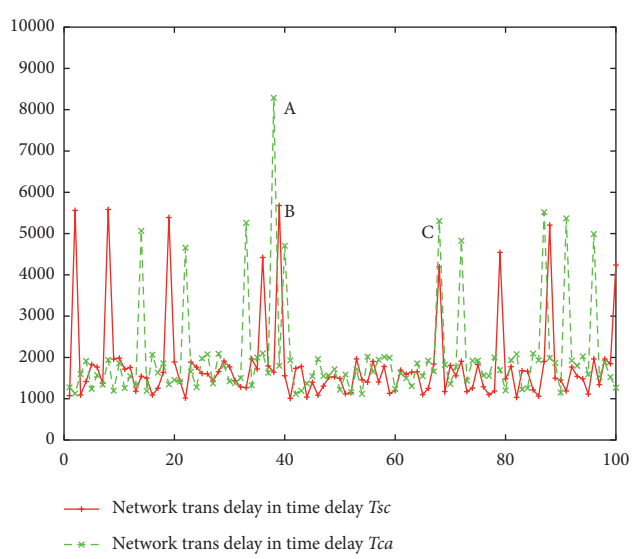


Figure 7: Network transmission latency in the HCPN model for NCSs.

Table 2: Main parameters.

Module	Time	
PLC module	IO scan cycle	17000
	Read	2000
	Execute	8000
	Write	2000
Ethernet module	2000	
Remote analog input/output	IO-read	1000
	IO-write	1000

$$A_{sc} = \begin{bmatrix} 0.28 & 0.72 \\ 0.84 & 0.15 \end{bmatrix},$$
$$B_{sc} = [B_{sc1} \quad B_{sc2}],$$
$$B_{sc1} = \begin{bmatrix} 0.15 & 0.10 & 0.15 & 0.15 & 0.15 & 0.13 & 0.17 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$B_{sc2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.20 & 0.20 & 0.18 & 0.20 & 0.12 & 0.03 & 0.04 & 0.03 \end{bmatrix}. \tag{25}$$

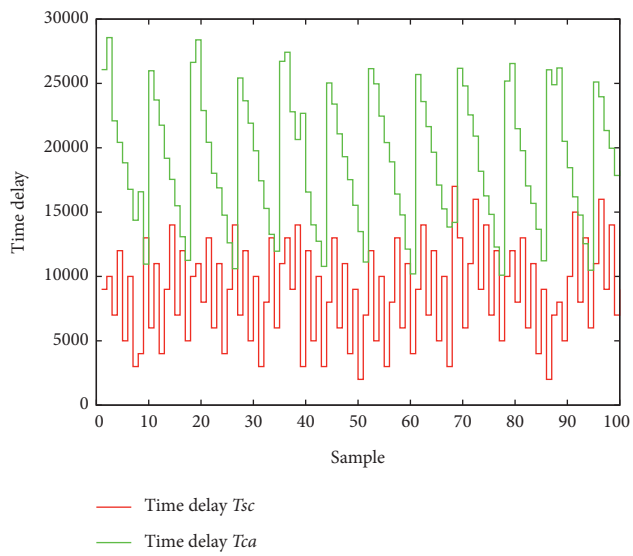


Figure 8: Tsc time delay and Tca time delay.

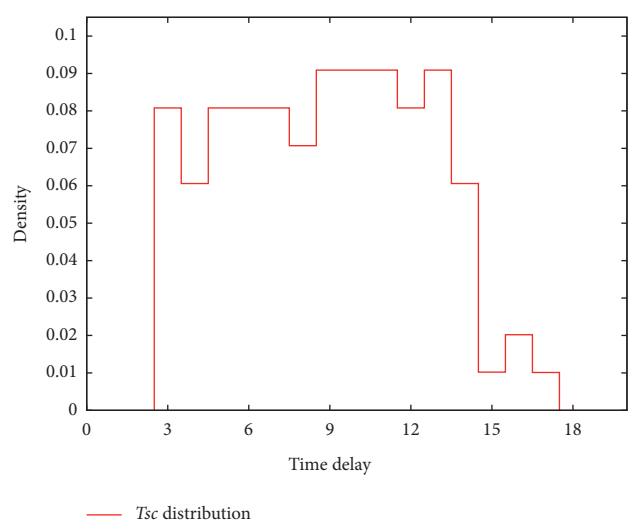


Figure 9: Distribution of *Tsc* in the HCPN model.

In the similar way, the system $\lambda_{ca} = \{A_{ca}, B_{ca}, \pi_{0_{ca}}\}$ with 2-state and 19-observations can be calculated with

$$A_{ca} = \begin{bmatrix} 0.78 & 0.22 \\ 0.25 & 0.75 \end{bmatrix},$$
$$B_{ca} = [\, B_{ca1} \quad B_{ca2} \,],$$
$$B_{ca1} = \begin{bmatrix} 0.06 & 0.10 & 0.08 & 0.11 & 0.11 & 0.11 & 0.08 & 0.19 & 0.10 & 0.06 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$B_{ca2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.19 & 0.12 & 0.10 & 0.15 & 0.06 & 0.15 & 0.15 & 0.06 & 0.02 \end{bmatrix}. \tag{26}$$

5.3. Validation Results. From the above section, the important matrixes can be obtained as

$$A = \{A_{sc}, A_{ca}\},$$
$$B = \{B_{sc}, B_{ca}\}. \tag{27}$$

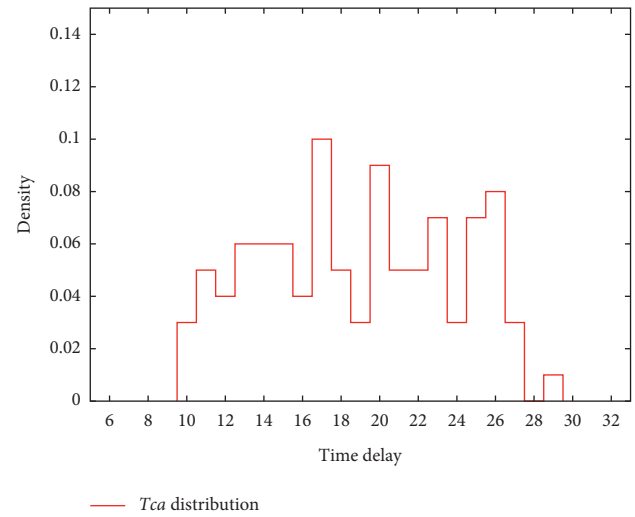


Figure 10: Distribution of *Tca* in the HCPN model.

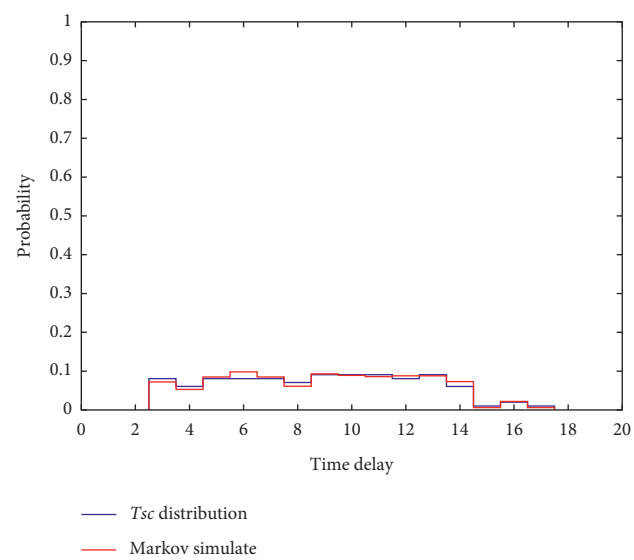


Figure 11: The comparison of *Tsc* probability distribution between the HCPN model and the Markov model.

Using the data from those two matrices, the stationary distribution of the Markov model λ_{sc} and λ_{ca} can be estimated by those two time delays (Tca and Tsc) with the Matlab statics toolbox. The probability density function (PDF) can be calculated by sample arrays which can be obtained from HCPN consecutive time delay sequence. The PDF is the distribution of the time delay which can be compared with the stationary distribution of the Markov model.

Figures 11 and 12 are the stationary distribution of the Hidden Markov Model building above, and every observation probability value is similar to the probability density function of the time-delay sequence. The figures show that the Markov stationary distribution of both the time delay Tsc and the time delay Tca are approximately equivalent to the probability density function of HCPN. It is clearly seen

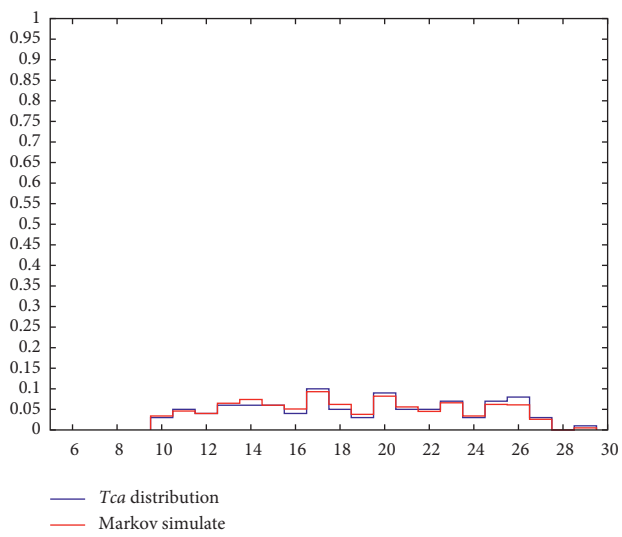


Figure 12: The comparison of *Tca* probability distribution between the HCPN model and the Markov model.

that building a Markov modeling of NCSs is feasible and time delays conform to Markov lain.

6. Conclusion

In this paper, we have investigated a novel HCPN model approach for the network control system with PLC. The modified structure-conserving model has been accomplished to calculate sequences of delays under the control of scan cycles for PLC-CPU. Besides, a series of special phenomenon has been taken into consideration based on the traditional mode, such as data packets drop and data packets out of order. Time delays in network transmission have been calculated to observe the transmission data packet state in Ethernet. Finally, Markov analytical models have been built for analyzing time delays which can be obtained from the forward step mathematically.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant nos. 61403278 and 61503280).

References

[1] H. Li, C. Wu, S. Yin, and H.-K. Lam, "Observer-based fuzzy control for nonlinear networked systems under unmeasurable premise variables," *IEEE Transactions on Fuzzy Systems*, vol. 24, pp. 1233–1245, 2015.

[2] H. Li, Z. Chen, L. Wu, H. K. Lam, and H. Du, "Event-triggered fault detection of nonlinear networked systems," *IEEE Transactions on Cybernetics*, vol. 47, pp. 1041–1052, 2016.

[3] S. Yin, X. Zhu, J. Qiu, and H. Gao, "State estimation in nonlinear system using sequential evolutionary filter," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3786–3794, 2016.

[4] H. Gao and X. Li, "Filtering for discrete-time state-delayed systems with finite frequency specifications," *IEEE Transactions on Automatic Control*, vol. 56, no. 12, pp. 2935–2941, 2011.

[5] H. Li, Y. Gao, P. Shi, and H.-K. Lam, "Observer-based fault detection for nonlinear systems with sensor fault and limited communication capacity," *IEEE Transactions on Automatic Control*, vol. 61, pp. 2745–2751, 2015.

[6] H. Li, C. Wu, L. Wu, H.-K. Lam, and Y. Gao, "Filtering of interval type-2 fuzzy systems with intermittent measurements," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 668–678, 2016.

[7] J. Zhang, C. Peng, D. Du, and M. Zheng, "Adaptive event-triggered communication scheme for networked control systems with randomly occurring nonlinearities and uncertainties," *Neurocomputing*, vol. 174, pp. 475–482, 2016.

[8] W. Wang, D. Nešić, and R. Postoyan, "Emulation-based stabilization of networked control systems implemented on flexray," *Automatica*, vol. 59, pp. 73–83, 2015.

[9] B. Liu, B. Qiu, Y. Cui, and J. Sun, "Fault-tolerant H ∞ control for networked control systems with randomly occurring missing measurements," *Neurocomputing*, vol. 175, pp. 459–465, 2016.

[10] G. P. Liu, "Design and analysis of networked non-linear predictive control systems," *IET Control Theory & Applications*, vol. 9, no. 11, pp. 1740–1745, 2015.

[11] X. Liu, X. Yu, G. Ma, and H. Xi, "On sliding mode control for networked control systems with semi-markovian switching and random sensor delays," *Information Sciences*, vol. 337-338, pp. 44–58, 2016.

[12] L. Schenato, "Optimal estimation in networked control systems subject to random delay and packet loss," in *Proceedings of the 45th IEEE Conference on Decision and Control, 2006*, pp. 5615–5620, IEEE, San Diego, CA, USA, December 2006.

[13] X.-S. Zhan, J. Wu, T. Jiang, and X.-W. Jiang, "Optimal performance of networked control systems under the packet dropouts and channel noise," *ISA Transactions*, vol. 58, pp. 214–221, 2015.

[14] D. Goswami, R. Schneider, and S. Chakraborty, "Re-engineering cyber-physical control applications for hybrid communication protocols," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, Paris, France, February 2011.

[15] M. Razeghi-Jahromi and A. Seyedi, "Stabilization of networked control systems with sparse observer-controller networks," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1686–1691, 2015.

[16] K. Liu, E. Fridman, and K. H. Johansson, "Dynamic quantization of uncertain linear networked control systems," *Automatica*, vol. 59, pp. 248–255, 2015.

[17] F. Cacace, F. Conte, A. Germani, and G. Palombo, "Optimal control of linear systems with large and variable input delays," *Systems & Control Letters*, vol. 89, pp. 1–7, 2016.

[18] L. Cao, H. Ren, W. Meng, H. Li, and R. Lu, "Distributed event triggering control for six-rotor UAV systems with asymmetric time-varying output constraints," *Science China Information Sciences*, vol. 64, no. 7, 2021.

[19] L. Qiu, Y. Shi, J. Pan, B. Xu, and H. Li, "Robust control for a networked direct-drive linear motion control system: design and experiments," *Information Sciences*, vol. 370-371, pp. 725–742, 2016.

[20] H. Arneson, N. Dousse, and C. Langbort, "A linear programming approach to routing control in networks of constrained nonlinear positive systems with concave flow rates," *Automatica*, vol. 68, pp. 357–368, 2016.

[21] H. Chen, J. Gao, T. Shi, and R. Lu, "H ∞ control for networked control systems with time delay, data packet dropout and disorder," *Neurocomputing*, vol. 179, pp. 211–218, 2016.

[22] X.-M. Sun, K.-Z. Liu, C. Wen, and W. Wang, "Predictive control of nonlinear continuous networked control systems with large time-varying transmission delays and transmission protocols," *Automatica*, vol. 64, pp. 76–85, 2016.

[23] S. Di Cairano, U. V. Kalabić, and I. V. Kolmanovsky, "Reference governor for network control systems subject to variable time-delay," *Automatica*, vol. 62, pp. 77–86, 2015.

[24] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall PTR, Hoboken, NJ, USA, 1981.

[25] R. S. Cost, Y. Chen, T. Finin, Y. Labrou, and Y. Peng, "Using colored petri nets for conversation modeling," *Issues in Agent Communication*, vol. 19, pp. 178–192, 2000.

[26] W. Chun-jian, L. Yong-zhi, and X. Fan, "An improved modeling method based on colored petri net," *Physics Procedia*, vol. 24, pp. 1128–1132, 2012.

[27] N. Gharbi, C. Dutheillet, and M. Ioualalen, "Colored stochastic petri nets for modelling and analysis of multiclass retrial systems," *Mathematical and Computer Modelling*, vol. 49, no. 7, pp. 1436–1448, 2009.

[28] R. Bastide and E. Barboni, "Software components: a formal semantics based on coloured petri nets," *Electronic Notes in Theoretical Computer Science*, vol. 160, pp. 57–73, 2006.

[29] M. Jamro, D. Rzonca, and W. Rząsa, "Testing communication tasks in distributed control systems with sysml and timed colored petri nets model," *Computers in Industry*, vol. 71, pp. 77–87, 2015.

[30] H. Huang, H. Kirchner, Secure Interoperation in Heterogeneous Systems Based on Colored Petri Nets, Working Paper or Preprint (2009). URLhttps://hal.inria.fr/inria-00396952.

[31] A. Ghanaim and G. Frey, "Component based colored petri net model for ethernet based networked control systems," in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1100–1103, Berlin, Germany, September 2008.

[32] A. Ghanaim, G. A. Borges, and G. Frey, "Estimating delays in networked control systems using colored petri nets and Markov chain models," in *Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–6, Berlin, Germany, September 2009.

[33] P. Huber, K. Jensen, and R. M. Shapiro, "Hierarchies in coloured petri nets," in *Advances in Petri Nets 1990*, pp. 313–341, Springer, Berlin, Germany, 1989.

[34] M. Elkoutbi and R. K. Keller, "Modeling interactive systems with hierarchical colored petri nets," in *Advanced Simulation Technologies Conference*, pp. 432–437, Citeseer, Princeton, NJ, USA, 1998.

[35] Y. Yang, Q. Tan, and Y. Xiao, "Verifying web services composition based on hierarchical colored petri nets," in *Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems*, pp. 47–54, ACM, Bremen, Germany, November 2005.

[36] A. Ghanaim and G. Frey, "Markov modeling of delays in networked automation and control systems using colored petri net models simulation," in *Proceedings of the 18th IFAC World Congress, Anonymous IFAC Secretariat*, pp. 2731–2736, Milano, Italy, August 2011.

[37] A. Ghanaim and G. Frey, "Modeling and control of closed-loop networked plc-systems," in *Proceedings of the American Control Conference (ACC)*, pp. 502–508, Milwaukee, Wisconsin, USA, June 2011.

[38] G. Mustafa and T. Chen, "H filtering for nonuniformly sampled systems: a markovian jump systems approach," *Systems & Control Letters*, vol. 60, no. 10, pp. 871–876, 2011.

[39] Á. Cuenca, U. Ojha, J. Salt, and M.-Y. Chow, "A non-uniform multi-rate control strategy for a Markov chain-driven networked control system," *Information Sciences*, vol. 321, pp. 31–47, 2015.

[40] Y. Wang, S. Xu, and S. Zhang, "Fault detection for a class of nonlinear networked control systems with markov sensors assignment and random transmission delays," *Journal of the Franklin Institute*, vol. 351, no. 10, pp. 4653–4671, 2014.

[41] H. Gao, Z. Fei, J. Lam, and B. Du, "Further results on exponential estimates of markovian jump," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, p. 223, 2011.

[42] K. Jensen, "Coloured petri nets," *EATCS Monographs on Theoretical Computer Science*, vol. 1.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 9963382, 8 pages
https://doi.org/10.1155/2021/9963382

Research Article

Application of an Improved CHI Feature Selection Algorithm

Liang-jing Cai,[1] **Shu Lv**,[1] **and Kai-bo Shi**[2]

[1]*School of Mathematical Sciences, University of Electronic Science and Technology of China, Sichuan, Chengdu 611731, China*
[2]*School of Electronic Information and Electrical Engineering, Chengdu University, Sichuan, Chengdu 610106, China*

Correspondence should be addressed to Shu Lv; lvshu@uestc.edu.cn

Received 11 March 2021; Accepted 4 May 2021; Published 13 May 2021

Academic Editor: Zi-Peng Wang

Text classification is the critical content of machine learning, and it is widely applied in information filtering, sentimental analysis, and text review. It is very important to improve the accuracy of classification results, and this is also the main research purpose of researchers in this field in recent years. Feature selection plays an important role in text classification, which has the functions of eliminating irrelevant features, reducing dimensionality, and improving classification accuracy. So, this paper studies the CHI feature selection algorithm, and the main work and innovations are as follows: firstly, this paper analyzed the CHI algorithm's flaws, determined that the introduction of new parameters will be the improvement direction of the CHI algorithm, and thus proposed a new algorithm based on variance and coefficient of variation. Secondly, experiment to verify the effectiveness of the new algorithm. In terms of language, the experiment in this paper includes two text classification systems, which were Chinese and English. In terms of classifiers, two classifier algorithms were used, which included the KNN classifier and the Naive Bayes classifier. In terms of data types, two distribution types of data were used: balanced datasets and unbalanced datasets. Finally, experiment and result analysis. This paper has conducted 3 comparative experiments and analyzed the results of each experiment. The experimental results obtained are all significantly improved compared to the results before the improvement.

1. Introduction

Text classification is a process of classifying unknown text according to the content of text [1]. It is the most classical field of natural language processing, which is according to the content of the text; using some automatic classification algorithms, the computer divides the texts into predefined categories [2]. With the rapid development of the Internet, information and data exist in a variety of ways, such as images, video, sound, and text. Compared with other forms, texts are applied more widely, because of its faster upload speed, faster download speed, and less use of network resources. Therefore, to quickly and correctly classify the massive text information, text classification technology has emerged. The processing of text categorization can be roughly divided into three stages: text preprocessing, text feature dimension reduction, and classification model construction. Text classification is widely used in filtering spam, analysis of Internet public opinion [3], and clinical diagnosis [4]. At present, the main direction of the text information structure is the vector model, in which the difficulty is mainly the high dimension of text features and the data sparsity [5], so feature dimension reduction plays a vital role in the classification effect. Feature selection method is an effective way to address this issue and play an important role in text classification [6].

At present, many researchers have proposed varieties of text feature selection methods, which include document frequency (DF), information gain (IG), mutual information (MI), expected cross-entropy (ECE), chi-square (CHI), and Gini Index. Many papers have proved that IG and CHI work best for text classification [7, 8]. Because the CHI statistical algorithm has many advantages, such as low complexity, easy understanding, and significant classification effect, people use it widely in actual situations. However, CHI algorithm only considers the influence of the document frequency in the calculation process, but ignores the factor of the feature word frequency; this defect is known as the "low word frequency defect"; many researchers have proposed improved methods for CHI algorithm. This paper also

proposes an improvement in the CHI algorithm based on the variance (Var) and coefficient of variation, which named Var-CV-CHI algorithm in this paper.

The rest of this paper is organized as follows: Section 2 briefly reviews the related work; Section 3 presents the method of improved CHI algorithm; Section 4 shows the experimental results and analyzes the results; Section 5 concludes the paper with some possible work in the future.

2. Related Work

Feature selection aims at removing irrelevant or redundant features for solving a supervised learning problem and reducing the difficulty of learning tasks [9]. It has been widely studied in web document processing (text classification, text retrieval, text recovery, etc.), gene analysis, drug diagnosis, and other fields.

Feature selection has a significant impact on the results of text classification, where it is often used to find the smallest subset of features that maximally increases the performance of the model. An excellent feature selection model can optimize the algorithm and improve the performance of text classification. Facing increasingly diverse forms of data or data flow, how to design better feature selection algorithms to solve the needs of society is a long-term task.

Machine learning divides feature selection into three methods: embedded, wrapper, and filter methods.

In the embedded feature selection method, some machine learning algorithms and models are first used for training to obtain the weight coefficients of each feature, and the features are selected from large to small according to the coefficients. The wrapper method selects or excludes several features at a time based on the objective function (usually the predictive effect score). The filter method scores each feature according to its divergence or correlation, sets the threshold, and selects the features. Filter-based feature selection is an effective solution to improve the performance of classification systems by selecting significant features and discarding the undesirable ones [10]. The features with a score higher than the threshold were selected, or the features with the most significant score in the first k scores were selected. Accurately, calculate the divergence of each feature, remove the features with divergence less than the threshold, and select the features with the most significant first k fractions; calculate the correlation between each feature and the label, remove the features whose correlation is less than the threshold, and select the feature with the most significant first k scores. The classical filter feature selection methods are variance threshold, chi-square, MI, etc. Moreover, chi-square performs best.

Many researchers propose improved methods based on the chi-square feature selection method. The related works are as follows.

Fan et al. [11] introduced word frequency factor, interclass variance, and adaptive scaling factor into CHI, which significantly improved the effect of text classification. Pei [12] introduced factors such as dispersion, concentration, and word frequency into the CHI formula, which improved its classification accuracy on the unevenly distributed corpus. Qiu et al. [13] proposed the variance-based chi-square statistics (Var-CHI) method which has significantly improved the recall and precision. Bahassine et al. [14] proposed an Imp-CHI algorithm, which combined the total number of documents in the corpus with the number of documents belonging to a particular category for Arabic text classification. However, they all did not consider using the variance (Var) of total word frequency in each class and coefficient of variation (CV) within a class to measure the dispersion of feature words in the category.

3. Materials and Methods

3.1. Chi-Square Feature Selection. The CHI algorithm's core idea is hypothesis testing; by observing the deviation between the actual value and the theoretical value, we can judge the correlation between the feature words and the category. In the text classification process, the original hypothesis is that "feature word t is not related to class C_j," and each feature word can get the corresponding CHI value. The larger the CHI value, the more significant the correlation between t and C_j. According to the order of CHI values from large to small, we can select m feature words that have the most significant correlation with class C_j.

The CHI value is defined as the following equation:

$$\chi^2(t, C_j) = \frac{N \times (AD - BC)^2}{(A + C)(B + D)(A + B)(C + D)}, \tag{1}$$

where N is the total number of documents in the corpus; A is the number of documents that contain feature t and belong to class; B is the number of documents that contain feature t but do not belong to class; C is the number of documents that do not contain feature t but belong to class; D is the number of documents that do not contain feature t and not belong to class.

According to equation (1), each feature word gets a score, and then the first m feature words are selected.

3.2. Improved Chi-Square Algorithm. Yang's research shows that chi-square is one of the best current feature selection methods [8]. Compared with other methods, the CHI method can reduce more vocabulary and eliminate a lot of redundant words and then improve the classification performance. Moreover, with the amount of text gradually increasing, the stability is also considerable. Since $A + C$ is equal to the number of documents belonging to category C_j, $B + D$ is equal to the number of documents not belonging to category C_j, both of which are constant and not related to the feature word t, so does N. Therefore, equation (1) can be written as the following equation:

$$\chi^2(t, C_j) = \frac{(AD - BC)^2}{(A + B)(C + D)}. \tag{2}$$

In equation (2), A, B, C, and D all come from the statistics of the number of documents containing feature word t_k in each category of the corpus, so the CHI value is only related to whether the feature words appear, but not related

to the frequency of its appearance. This problem leads to incomplete classification information; it is easy to select low-frequency words that are not useful for classification falsely. Especially when the distribution of corpus categories is uneven, equation (2) is more likely to give high scores to low-frequency words, which affects the accuracy of classification results. To correct these shortcomings of the traditional CHI algorithm, this paper proposes an improved algorithm, Var-CV-CHI algorithm. The method is as follows:

(a) In the text classification, we want to select feature words that meet the condition of $AD - BC > 0$ as many as possible. However, according to equation (2), the feature words satisfying $AD - BC < 0$ will be selected at the same time, so the low-frequency words in the category will be selected by mistake; we call it as a negative correlation. Some low-frequency words contain important classification information, so we cannot completely delete them. Therefore, this paper will select feature words according to the positive and negative correlation between feature words and categories and give them different weights to reduce the negative correlation of low-frequency words. The positive correlation and negative correlation are as shown in the following equations:

$$\chi^2\left(t, C_j\right)^+ = \frac{(AD - BC)^2}{(A + B)(C + D)}, \quad AD - BC > 0, \tag{3}$$

$$\chi^2\left(t, C_j\right)^- = \frac{(AD - BC)^2}{(A + B)(C + D)}, \quad AD - BC < 0. \tag{4}$$

Thus, add an adaptive scaling factor α. The equation is improved as the following equation:

$$\chi^2\left(t, C_j\right) = \alpha \times \chi^2\left(t, C_j\right)^+ + (1 - \alpha) \times \chi^2\left(t, C_j\right)^-, \tag{5}$$

($\alpha \in (0.5, 1)$. In the experiment of this paper, $\alpha$is 0.8).

(b) As described above, in the original CHI equation, it does not count t_k's frequency. When multiclass texts are classified, the larger the number of documents containing the feature word t_k and t_k's frequency in category C_j, the better the t_k's classification ability. Therefore, this paper proposes these parameters as the following equations:

$$\beta = \frac{A^2}{(B + 1)}, \tag{6}$$

$$\gamma = \frac{\sum_{i=1}^{n} tf\left(t_k, d_{ij}\right)}{\sum_{j=1}^{r}\sum_{i=1}^{n} tf\left(t_k, d_{ij}\right) + 1}, \tag{7}$$

where $tf(t_k, d_{ij})$ is the number of t_k in the ith file in class C_j, r is the total number of news categories, and n is the total number of documents in a category.

(c) If feature words t_k have good classification ability, they should mostly appear in one category. On the contrary, they seldom or even do not appear in other categories. Therefore, the greater the dispersion of t_k's frequency between categories, that is, the larger the variance between categories, the stronger the classification ability of the feature words t_k. Thus, this paper gives a parameter ξ_1 as the following equation:

$$\xi_1 = \left| \sum_{i=1}^{n} tf\left(t_k, d_{ij}\right) - \frac{\sum_{j=1}^{r}\sum_{i=1}^{n} tf\left(t_k, d_{ij}\right)}{r} \right|^2. \tag{8}$$

$tf(t_k, d_{ij})$'s meaning is the same as equation (7); $\sum_{j=1}^{r}\sum_{i=1}^{n} tf(t_k, d_{ij})/r$ represents the category average word frequency of t_k in the corpus. If $\sum_{i=1}^{n} tf(t_k, d_{ij}) - (\sum_{j=1}^{r}\sum_{j=1}^{n} f(t_k, d_{ij})/r) \geq 0$, it indicates that the word frequency of t_k in the specified category C_j is greater than or equal to the average word frequency of t_k between categories, and it is a meaningful word for C_j; if $\sum_{i=1}^{n} tf(t_k, d_{ij}) - (\sum_{j=1}^{r}\sum_{i=1}^{n} tf(t_k, d_{ij})/r) < 0$, it indicates that the word frequency of t_k in the specified category C_j is less than the average word frequency of t_k between categories, and it is a meaningless word for C_j. According to the mathematical meaning of ξ_1, $\xi_1 \geq 0$ is constantly established; so, if ξ_1 is very large and $\sum_{i=1}^{n} tf(t_k, d_{ij}) - (\sum_{j=1}^{r}\sum_{i=1}^{n} tf(t_k, d_{ij})/r) > 0$, then it shows that t_k has important classification significance to C_j. In order for the formula to be neat and beautiful, command $\sum_{i=1}^{n} tf(t_k, d_{ij}) - (\sum_{j=1}^{r}\sum_{i=1}^{n} tf(t_k, d_{ij})/r) = s$, and adjust parameter ξ_1 to ξ. The value of ξ is shown as the following equation:

$$\xi = \left\{ \sum_{i=1}^{n} tf\left(t_k, d_{ij}\right) - \sum_{j=1}^{r}\sum_{i=1}^{n} tf\left(t_k, d_{ij}\right), s \geq 0, 0, s \leq 0. \right. \tag{9}$$

(d) Coefficient of variation is a normalization index of the degree of dispersion of a probability distribution, which is defined as the following equation:

$$\mathrm{CV} = \frac{\sigma}{\bar{x}} \times 100\%. \tag{10}$$

In the text classification, feature words are distributed in every text of a category as evenly as possible, which indicates that it has a strong classification ability. The smaller the coefficient of variation, the more uniform the data distribution. To make the improved CHI value proportional to each parameter, we reverse the numerator and denominator of the coefficient of variation and then transform the standard deviation into a variance. Therefore, we introduce the coefficient of variation for chi-square, which is defined as the following equation:

$$\varepsilon = \left(\frac{\mu}{\sigma^2 + 1}\right) = \frac{\sum_{i=1}^{n} tf\left(t_k, d_{ij}\right)/n}{\left|tf\left(t_k, d_{ij}\right) - \left(\sum_{i=1}^{n} tf\left(t_k, d_{ij}\right)/n\right)\right|^2 + 1}. \tag{11}$$

In summary, the feature selection segment should select words that appear concentrated in a specific category and are evenly distributed in that category. At the same time, the more frequently these words are in this specific category, the better their category representativeness. This paper proposes an improved CHI algorithm, which is called Var-CV-CHI algorithm; the calculation method is as the following equation:

$$\chi^2\left(t_k, C_j\right) = \beta \times \gamma \times \xi \times \varepsilon \times \left[\alpha \times \chi^2\left(t_k, C_j\right)^+ + (1-\alpha) \times \chi^2\left(t_k, C_j\right)^-\right]. \tag{12}$$

4. Results and Discussion

In this experiment, Chinese news corpus and English news corpus are both used. Because of the different language structures of Chinese and English, we construct Chinese text classification system and English text classification system, respectively. The specific experimental settings and results are as follows.

4.1. Data Collection

4.1.1. Chinese News Corpus. The Chinese corpus is from School of Information Management, Sun Yat-sen University. The corpus is a manually labeled corpus with 14 categories and a total of 36865 documents with high accuracy. In order to make the sample size large enough, this paper selects 10 categories with a large amount of text. The experiments use two corpus sets: balanced corpus (Table 1) and unbalanced corpus (Table 2); same 10 categories are selected in total.

4.1.2. English News Corpus. The English corpus is the Reuters-21578 News Corpus. According to the preliminary experiment results, because this corpus is skewed data, the effect of the Var-CV-CHI algorithm on the English balanced corpus is not significantly improved compared with the traditional CHI algorithm, so only the unbalanced corpus is used in this experiment (Table 3).

4.2. Experiment. In this paper, the Var-CV-CHI algorithm is compared with the traditional CHI and other improved algorithms on the balanced corpus or the unbalanced corpus. To exclude the influence of the classifier, we choose to use two kinds of classifiers for experiments, KNN classifier and Naïve Bayes classifier.

Figure 1 shows the experimental process.

In the experiments, the ratio of the train set and test set is 4 : 1. The purpose of CHI feature selection is to select the first m feature words based on the calculated CHI value. According to the size of the dataset, the threshold value of feature words selected from each category is 150 in Chinese corpus and 20 in English corpus. To verify the effectiveness and stability of the Var-CV-CHI algorithm, we set up the following experiments.

Table 1: The number of Chinese balanced corpora.

Category	Number
Culture	1500
Education	1500
Entertainment	1500
Event	1500
Finance	1500
Game	1500
Health	1500
Occultism	1500
Sport	1500
Technology	1500

Table 2: The number of Chinese unbalanced corpora.

Category	Number
Culture	150
Education	610
Entertainment	1050
Event	800
Finance	1200
Game	840
Health	300
Occultism	300
Sport	700
Technology	1500

Table 3: The number of English unbalanced corpora.

Category	Number
Acq	460
Crude	80
Earn	645
Interest	218
Ship	95
Student	105
Trade	300

4.2.1. Experimental in Chinese Corpus

(1) Experiment 1. Tables 4 and 5 show the experimental results of a balanced corpus and unbalanced corpus in the KNN classifier, which contains a precision, recall, and F1-score.

After the contrast experiment, it is found that when $k = 7$, the classification effect on the KNN classifier is the best. Figure 2 shows the F1 value of the classifier when k takes different values.

From Tables 4 and 5, compared with the traditional CHI algorithm, even if the performance of a few categories decline, for example, the F1-value of education and event on the unbalanced corpus drops by 2%, but does not affect the improvement of the overall corpus classification performance. In a word, whether in the balanced corpus or the unbalanced corpus, the Var-CV-CHI algorithm performed

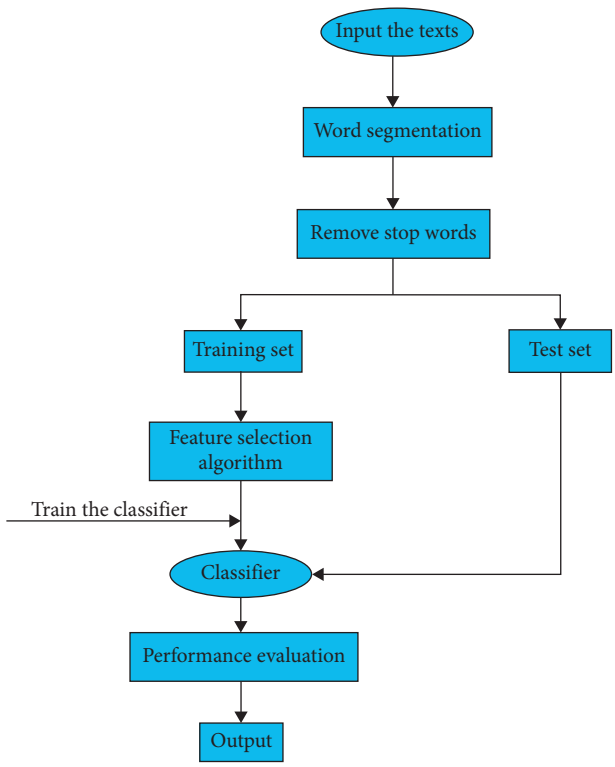


Figure 1: The process of feature selection in text classification.

Table 4: Comparison of classification results between CHI and Var-CV-CHI on the balanced Chinese corpus (KNN).

Category	CHI			Var-CV-CHI		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Culture	0.92	0.95	0.93	0.94	0.94	0.94
Education	0.94	0.93	0.94	0.93	0.92	0.92
Entertainment	0.94	0.88	0.91	0.93	0.87	0.90
Event	0.82	0.82	0.82	0.75	0.85	0.80
Finance	0.82	0.88	0.85	0.87	0.89	0.88
Game	0.98	0.97	0.98	0.98	0.97	0.97
Health	0.92	0.78	0.84	0.95	0.82	0.88
Occultism	0.77	0.98	0.86	0.83	1.00	0.90
Sport	0.98	0.92	0.95	0.98	0.91	0.95
Technology	0.92	0.84	0.88	0.93	0.86	0.89
Macro-avg-P		0.90			0.91	
Macro-avg-R		0.90			0.90	
Macro-avg-F1		0.90			0.90	

well. Macro-avg-P and macro-avg-F1 are 3% and 2%, respectively, on the unbalanced corpus. The traditional CHI algorithm performs better on the balanced corpus, while the Var-CV-CHI algorithm performs well on both the balanced corpus and the unbalanced corpus. Figures 3 and 4 show the comparison of the macroresults of CHI and Var-CV-CHI.

(2) Experiment 2. Tables 6 and 7 show the experimental results of the balanced corpus and unbalanced corpus in the Naïve Bayes classifier.

From Tables 6 and 7, the results on the unbalanced corpus are better than the balanced corpus, in which macro-avg-P, macro-avg-R, and macro-avg-F1 are increased by 8%,

Table 5: Comparison of classification results between CHI and Var-CV-CHI on the unbalanced Chinese corpus (KNN).

Category	CHI			Var-CV-CHI		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Culture	0.97	0.90	0.93	0.93	0.92	0.92
Education	0.93	0.82	0.87	0.93	0.87	0.90
Entertainment	0.92	0.91	0.91	0.91	0.93	0.92
Event	0.79	0.80	0.80	0.79	0.80	0.80
Finance	0.86	0.84	0.85	0.88	0.88	0.88
Game	0.97	0.97	0.97	0.97	0.98	0.97
Health	0.51	0.92	0.66	1.00	0.68	0.81
Occultism	0.64	1.00	0.78	0.47	1.00	0.64
Sport	0.97	0.86	0.91	0.96	0.84	0.90
Technology	0.93	0.79	0.86	0.94	0.87	0.91
Macro-avg-P		0.85			0.88	
Macro-avg-R		0.88			0.88	
Macro-avg-F1		0.85			0.87	

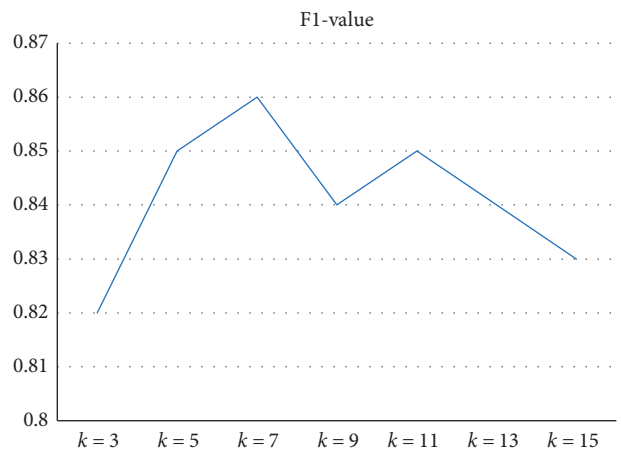


Figure 2: Classification effect under different k values.

6%, and 9%, respectively; it is a significant improvement compared with traditional CHI algorithm. The experiments show a fact: the classification performance of a balanced corpus is better on the KNN classifier, while an unbalanced corpus is better on the Naïve Bayes classifier. The Var-CV-CHI algorithm is better than the CHI algorithm in any situation and shows excellent performance. Figures 5 and 6 show the comparison of the macroresults of CHI and Var-CV-CHI.

4.2.2. Experimental in English Corpus

(1) Experiment 3. Tables 8 and 9 show the experimental results of the English unbalanced corpus using the Naïve Bayes classifier and KNN classifier.

From Tables 8 and 9, it can be seen that the classification results of the Var-CV-CHI are significantly improved in both KNN classifier and Naïve Bayes classifier, and it works best on the KNN classifier, in which macro-avg-P, macro-avg-R, and macro-avg-F1 are increased by 2%, 7%, 5%, respectively. On the Naïve Bayes classifier, the category of

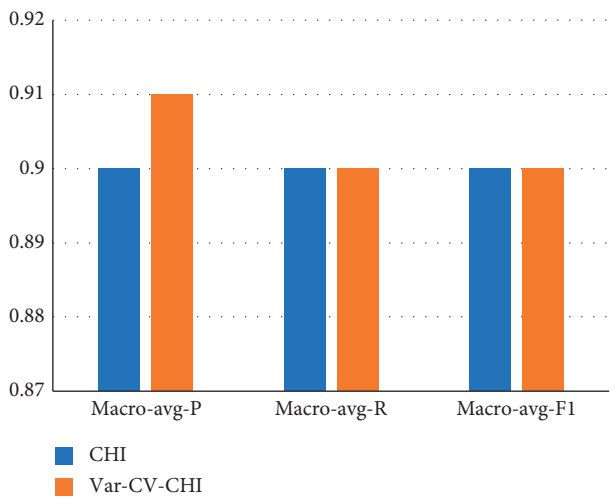


Figure 3: Comparison of macroresults between CHI and Var-CV-CHI on the balanced Chinese corpus (KNN).

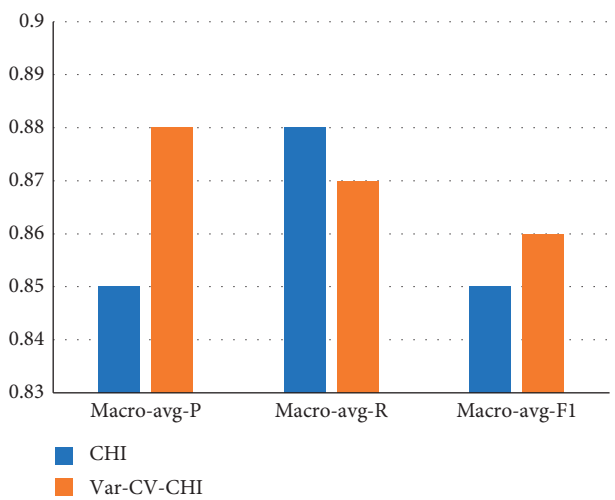


Figure 4: Comparison of macroresults between CHI and Var-CV-CHI on the unbalanced Chinese corpus (KNN).

Table 6: Comparison of classification results between CHI and Var-CV-CHI on the balanced Chinese corpus (Naïve Bayes).

Category	CHI			Var-CV-CHI		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Culture	0.99	0.57	0.72	0.99	0.56	0.72
Education	0.92	0.89	0.91	0.90	0.94	0.92
Entertainment	0.71	0.95	0.82	0.72	0.94	0.82
Event	0.65	0.83	0.73	0.73	0.84	0.78
Finance	0.83	0.73	0.78	0.83	0.82	0.83
Game	0.98	0.94	0.96	0.98	0.94	0.96
Health	0.95	0.91	0.93	0.95	0.93	0.94
Occultism	0.92	0.99	0.96	0.97	1.00	0.98
Sport	0.97	0.97	0.97	0.98	0.97	0.97
Technology	0.88	0.86	0.87	0.89	0.88	0.89
Macro-avg-P	0.88			0.89		
Macro-avg-R	0.86			0.88		
Macro-avg-F1	0.86			0.88		

Table 7: Comparison of classification results between CHI and Var-CV-CHI on the unbalanced Chinese corpus (Naïve Bayes).

Category	CHI			Var-CV-CHI		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Culture	0.96	0.84	0.89	0.96	0.84	0.90
Education	0.75	0.88	0.81	0.92	0.98	0.95
Entertainment	0.94	0.89	0.91	0.93	0.94	0.94
Event	0.68	0.62	0.65	0.73	0.80	0.77
Finance	0.85	0.69	0.76	0.86	0.83	0.85
Game	0.97	0.95	0.96	0.98	0.95	0.96
Health	0.38	1.00	0.55	0.87	1.00	0.93
Occultism	0.92	1.00	0.96	0.95	1.00	0.98
Sport	0.95	0.88	0.92	0.96	0.94	0.95
Technology	0.93	0.83	0.88	0.94	0.90	0.92
Macro-avg-P	0.83			0.91		
Macro-avg-R	0.86			0.92		
Macro-avg-F1	0.83			0.92		

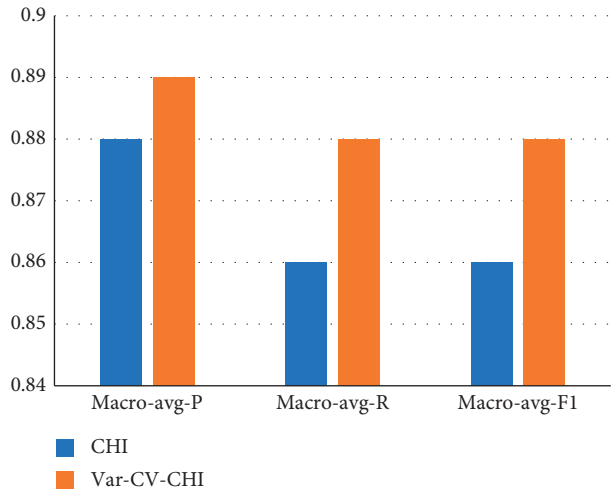


Figure 5: Comparison of macroresults between CHI and Var-CV-CHI on the balanced Chinese corpus (Naïve Bayes).

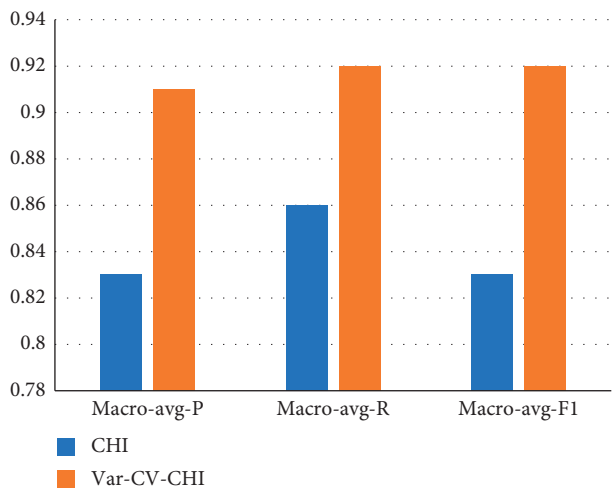


Figure 6: Comparison of macroresults between CHI and Var-CV-CHI on the unbalanced Chinese corpus (Naïve Bayes).

Table 8: Comparison of classification results between CHI and Var-CV-CHI on the unbalanced English corpus (KNN).

Category	CHI			Var-CV-CHI		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Acq	0.73	0.86	0.79	0.73	0.85	0.79
Crude	0.83	0.59	0.69	0.89	0.94	0.91
Earn	0.87	0.89	0.88	0.90	0.87	0.88
Interest	0.90	0.82	0.86	0.93	0.86	0.89
Ship	0.92	0.55	0.69	0.94	0.75	0.83
Student	1.00	1.00	1.00	1.00	0.91	0.95
Trade	0.93	0.89	0.91	0.95	0.90	0.92
Macro-avg-P	0.88			0.90		
Macro-avg-R	0.80			0.87		
Macro-avg-F1	0.83			0.88		

Table 9: Comparison of classification results between CHI and Var-CV-CHI on the unbalanced English corpus (Naïve Bayes).

Category	CHI			Var-CV-CHI		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Acq	0.76	0.84	0.80	0.80	0.84	0.82
Crude	0.78	0.82	0.80	0.84	0.94	0.89
Earn	0.93	0.85	0.89	0.93	0.88	0.90
Interest	0.95	0.91	0.93	0.93	0.93	0.93
Ship	0.85	1.00	0.92	0.92	1.00	0.96
Student	0.73	0.80	0.76	0.88	0.75	0.81
Trade	0.93	0.87	0.89	0.92	0.95	0.94
Macro-avg-P	0.85			0.89		
Macro-avg-R	0.87			0.90		
Macro-avg-F1	0.86			0.89		

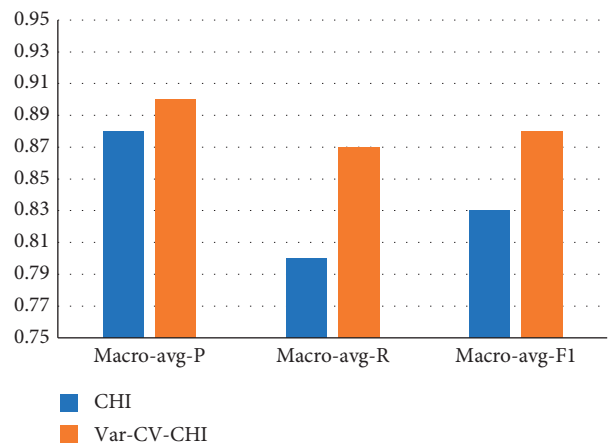


Figure 7: Comparison of macroresults between CHI and Var-CV-CHI on the unbalanced English corpus (KNN).

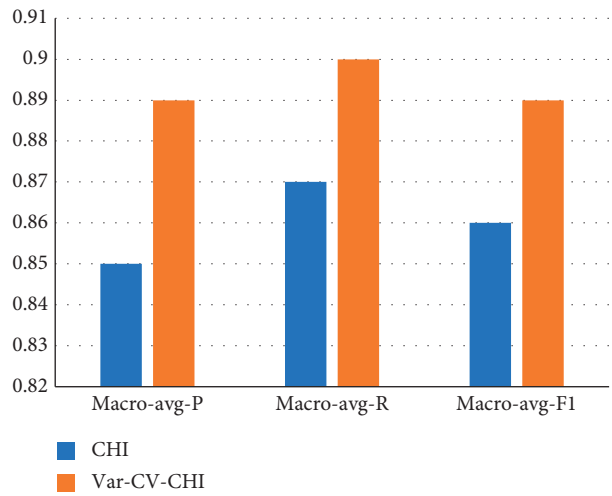


Figure 8: Comparison of macroresults between CHI and Var-CV-CHI on the unbalanced English corpus (Naïve Bayes).

Crude's classification effect of the Var-CV-CHI algorithm is especially improved significantly, and the P, R, and F are increased by 6%, 24%, and 18%, respectively. It can be seen from Table 3 that the number of files in the Crude category is the least among the 7 categories, so it can be seen that in the unbalanced corpus, the Var-CV-CHI's classification effect on the categories with the least number of files is the most obvious. In addition, through multiple experiments, we found that the more the uneven number of categories in the corpus, the better the classification results of the Var-CV-CHI algorithm. Figures 7 and 8 show the comparison of the macro-average results of CHI and Var-CV-CHI.

5. Conclusion

In this paper, through a lot of literature studies, and on this basis, it is determined that the introduction of new parameters will be the direction of CHI algorithm improvement, thus the Var-CV-CHI feature selection algorithm based on variance and coefficient of variation is proposed. Although many of the improved methods of the CHI algorithm are based on the idea of introducing new parameters, the method in this paper is more comprehensive and rigorous than these methods. It combines the differences caused by the different distributions of feature words between categories and within categories and synthesizes the situation of feature words in the document. The classification effect is very excellent. At the same time, compared with the existing works, the experiments in this paper are more abundant. Not only the language factor but also the distributions of classifiers and datasets are considered. Finally, the experiments from multiple angles proved the effectiveness of the Var-CV-CHI algorithm.

Text classification is a complex project, and every segment of it plays a vital role in the classification results. Future work can continue in the following directions:

The Var-CV-CHI algorithm in this paper only considers the distribution of feature words but without considering its semantic information. If the semantic information of feature items can be added to the feature selection algorithm, for

example, the synonyms with the same semantic information can be merged, and the feature words with the largest weight among the words with similar semantic information can be selected. Then, the redundancy of feature space will be reduced, and the performance of feature selection will be improved significantly.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The Chinese corpus of this article was provided by Lu Yonghe from the School of Information Management, Sun Yat-sen University.

References

[1] A. K. Uysal, "On two-stage feature selection methods for text classification," *IEEE Access*, vol. 6, pp. 43233–43251, 2018.

[2] Q. Zhou, M. S. Zhao, and H. U. Min, "Study on feature selection in Chinese text categorization," *Journal of Chinese Information Processing*, vol. 18, no. 3, pp. 17–23, 2004.

[3] Y. Zhao, S. Cheng, X. Yu, and H. Xu, "Chinese public's attention to the COVID-19 epidemic on social media: observational descriptive study," *Journal of Medical Internet Research*, vol. 22, no. 5, 2020.

[4] M. Oleynik, A. Kugic, Z. Kasáč, and M. Kreuzthaler, "Evaluating shallow and deep learning strategies for the 2018 n2c2 shared task on clinical text classification," *Journal of the American Medical Informatics Association*, vol. 26, no. 11, pp. 1247–1254, 2019.

[5] H. Liu, S. U. Zhan, and S. Liu, "Improved CHI text feature selection based on word frequency information," *Computer Engineering and Applications*, vol. 49, no. 22, pp. 110–114, 2013.

[6] X.-T. Wang and X.-Z. Luan, "Bayesian penalized method for streaming feature selection," *IEEE Access*, vol. 7, no. 99, pp. 103815–103822, 2019.

[7] H. Ji, A. H. Tan, and C. L. Tan, "On machine learning methods for Chinese document categorization," *Applied Intelligence*, vol. 18, no. 3, pp. 311–322, 2003.

[8] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, vol. 39, pp. 42–49, New York, NY, USA, August 1999.

[9] B. Li, "Importance weighted feature selection strategy for text classification," in *Proceeding of the 2016 International Conference on Asian Language Processing (IALP)*, Tainan, Taiwan, November 2016.

[10] O. A. M. Salem, F. Liu, Y.-P. P. Chen, and X. Chen, "Ensemble fuzzy feature selection based on relevancy, redundancy, and dependency," *Entropy*, vol. 22, no. 7, 2020.

[11] C. J. Fan, Y. S. Wang, and Y. T. Wang, "An improved CHI text feature selection algorithm," *Computer and Modernization*, vol. 11, pp. 7–11, 2016.

[12] Y. Pei, "Study on improved CHI for feature selection in Chinese text categorization," *Computer Engineering and Applications*, vol. 47, pp. 128–123, 2011.

[13] Y. F. Qiu, W. Wang, and D. Y. Liu, "CHI feature selection method based on variance," *Application Research of Computers*, vol. 29, pp. 1304–1306, 2012.

[14] S. Bahassine, A. Madani, and M. Kissi, "An improved Chi-sqaure feature selection for Arabic text classification using decision tree," in *Proceeding of the International Conference on Intelligent Systems: Theories & Applications*, Mohammedia, Morocco, October 2016.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 5577715, 8 pages
https://doi.org/10.1155/2021/5577715

Research Article

Simulation and Computational Study of CFD on Tube MBR Membrane Assembly

Kun Feng, Chunqing Li, Ming Zhang, and Xuting Liu

School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

Correspondence should be addressed to Chunqing Li; franklcq@163.com

Received 22 January 2021; Accepted 29 April 2021; Published 13 May 2021

Academic Editor: Zi-Peng Wang

When using tubular MBR to treat sewage, the water production is an important parameter to measure the efficiency of the tubular MBR system. The problem to be solved in this paper is to calculate the water yield of the tubular MBR system, so as to evaluate the sewage treatment efficiency of the MBR system. This research uses the CFD simulation software ANSYS 16.0 to study the water yield of the tubular MBR system. The MBR model of a single membrane filament tube was established using the ICEM CFD preprocessor in ANSYS 16.0, and the structured grid was divided to obtain a grid file. Then, the fluid solver was used to solve the mesh file and through the flow monitoring window to obtain the water output of the tubular MBR system. Finally, the CFD postprocessor in ANSYS 16.0 was used to visualize the calculation results and compare them with the waste-water treatment results of some actual MBR systems. The results show that the water yield calculated by the fluent solver is basically the same as that of the actual MBR system. This research realizes the purpose of calculating the water yield of the tubular MBR system with CFD technology, solves the problem of evaluating the working efficiency of the tubular MBR system with water consumption, and realizes the MBR before deployment The evaluation of the working efficiency of the system has certain reference value for the planning, design, and deployment of MBR.

1. Introduction

Membrane bioreactor (MBR) is a combination of membrane separation technology and biological treatment technology. It has the advantages of stable effluent quality, small floor area, and easy maintenance and management. The tubular MBR membrane module is one of the commonly used membrane modules in the MBR system, so this paper analyzes the tubular MBR membrane module. In the tubular MBR membrane module, the deployment of ceramic membrane will affect the working efficiency of the tubular MBR membrane module, and the water yield is an important parameter to measure the working efficiency of the tubular MBR membrane module, so the problem to be solved in this paper is how to calculate the water yield of the tubular MBR membrane module.

In order to calculate the water yield of tubular MBR membrane module, this paper uses the computational fluid dynamics (CFD) simulation software ANSYS 16.0 to model and calculate the tubular MBR membrane module. ANSYS 16.0 is mainly composed of the ICEM CFD preprocessor, FLUENT solver, and CFD-postprocessor. In the research process, firstly, the geometry model of a single membrane fiber tube MBR membrane module is built with ICEM CFD preprocessor, and the mesh of the geometry model is divided into fine meshes to generate mesh files. Then, the grid file is solved by fluent solver. Because the research object of this paper is to calculate the water yield of MBR membrane module, a flow monitoring window is set at the outlet of the model to observe the water yield. Finally, the CFD post-processor is used to visualize the calculation results. On this basis, in order to verify the correctness and reliability of fluent calculation results, this paper compares the simulation results with the actual MBR processing results.

2. Tubular MBR Membrane Module

The basic structure of the tubular MBR membrane module is shown in Figure 1 [1]. The tubular MBR membrane module

is mainly composed of a ceramic membrane and component housing. There are many small holes on the wall of the ceramic membrane, and these small holes are used to filter out suspended particles in the sewage [2]. When the tubular MBR system is operated, under the action of the water pump, the waste water which has been pretreated flows in from one end of the ceramic membrane and the filtered waste water flows from the small holes of the ceramic membrane wall to the component housing [3]. Finally, the staff can collect the filtered waste water at the outlet of the component housing. The concentrate formed by the solid suspension in the waste water flows out from the other end of the ceramic membrane. Compared with the traditional curtain MBR membrane module, the tubular MBR membrane module has the advantages of simple structure, strong adaptability, convenient cleaning, high pressure resistance, suitable for treating high viscosity, and high solid content liquid [4]. Therefore, the tubular MBR membrane modules are widely used in MBR waste-water treatment, and this paper uses CFD technology to simulate and calculate the tubular MBR membrane module [5].

3. CFD Simulation Software ANSYS 16.0

Computational fluid dynamics (CFD) is an interdisciplinary in fluid mechanics, mathematics, and computer science [6]. It can simulate and analyze various problems in practical engineering to solve various practical problems [7]. At present, there are many kinds of software for CFD simulation on the market, and ANSYS 16.0 is one of the commonly used CFD simulation software. ANSYS16.0 is mainly composed of ICEM CFD preprocessor, FLUENT solver, and CFD postprocessor. The main function of the preprocessor is to create geometric models and finely mesh the geometric models [8]. The main function of the solver is to calculate the mesh file, and the solution process is shown in Figure 2. The main function of the postprocessor is to present the calculation result of the solver in a graphical or image manner so that the user can more intuitively understand the CFD calculation result. In this paper, the CFD simulation software ANSYS 16.0 is used to simulate and calculate the tubular MBR membrane module.

4. Simulation of CFD on Tubular MBR

4.1. Preprocessor (ICEM CFD) Establishes a Tubular MBR Geometry Model. The tubular MBR geometric model of a single membrane tube established by ICEM CFD preprocessor is shown in Figure 3, and there are many small holes on the wall of the ceramic membrane [9]. Due to the limited effect of the preprocessor opening small holes, this paper sets the porous boundary conditions in the boundary conditions of the solver. During the operation of the model, the pretreated waste water flows into the ceramic membrane from the inlet and the sewage filtered through the ceramic membrane flows from the small holes of the wall surface of the ceramic membrane to the outlet of the component housing. Due to the entrapment of the ceramic tube wall, the concentrate composed of a large molecule particle or the like flows out from the other end of the ceramic film [10]. The following is a structured meshing of the geometric model.

Since the FLUENT solver is solved in units of grids, the geometry model needs to be meshed in the preprocessor. In ICEM CFD, there are two types of meshing: structured meshing and unstructured meshing. Structured meshing usually divides the geometric model into several quadrilaterals or hexahedrons. Unstructured meshing usually divides the geometric model into several triangles or tetrahedrons. In structured meshing, the determinant $2 \times 2 \times 2$ mesh quality map can measure the quality of the meshing. The principle of determinant $2 \times 2 \times 2$ mesh quality is to characterize the deformation of the unit by calculating the Jacobian determinant value of each hexahedron and then normalizing the matrix of the determinant. The value of one indicates an ideal hexahedral cube, and the value of zero indicates an inverse cube with a negative volume. The mesh quality is represented by the x-axis and all cells are between 0 and 1. If the value of a cell determinant is zero, this cube has one or more degenerate edges. In general, the determinant value above 0.3 can be accepted by most solvers. Since the mass distribution of the ceramic membrane is uniform, the paper uses structured mesh division, in which the O-shaped segmentation operation is performed on the cylindrical part of the geometric model. Finally, the resulting mesh file is shown in Figures 4 and 5 and is a determinant $2 \times 2 \times 2$ mesh quality map with a minimum mesh mass of 0.621. Therefore, the meshing of the geometric model is correct. Then, save the mesh file and prepare to solve the mesh file.

4.2. FLUENT Solver Calculates Mesh File. In this section, the mesh file in Section 4.1 is imported into the solver FLUENT, and then, the solution model description and solution process control operations are, respectively, performed in the FLUENT operation tree [11]. In solving the model description, the first step is to check the mesh file to ensure that the minimum cell volume is not negative and select a pressure-based steady-state solver and set the gravitational acceleration. The second step is selecting the basic physical model. Since the flow rate of the sewage in the tubular MBR system is low and the Reynolds number is lower than 2300, the flow state is set to the laminar flow model. The third step is to introduce fluid material from the material library and use water as the fluid material. The fourth step is setting the calculation area properties and sets the water as the calculation domain working medium. The fifth step is setting the calculation domain boundary conditions. First, the "Mass-Flow Inlet" is set as the inlet boundary condition and enters the water intake and operating pressure of the tubular MBR in "Mass-Flow Inlet" (these two parameters are provided by the sewage treatment plant.). Then, set "Outflow" to the outlet boundary condition. Finally, the "porous jump" is set to the wall boundary condition of the ceramic membrane, that is, many small holes are opened on the wall surface of the ceramic membrane. After these operations are completed, this paper will solve the process control operation.

In the solution process control, first, we set the solution method. In the drop-down list of pressure-velocity coupling

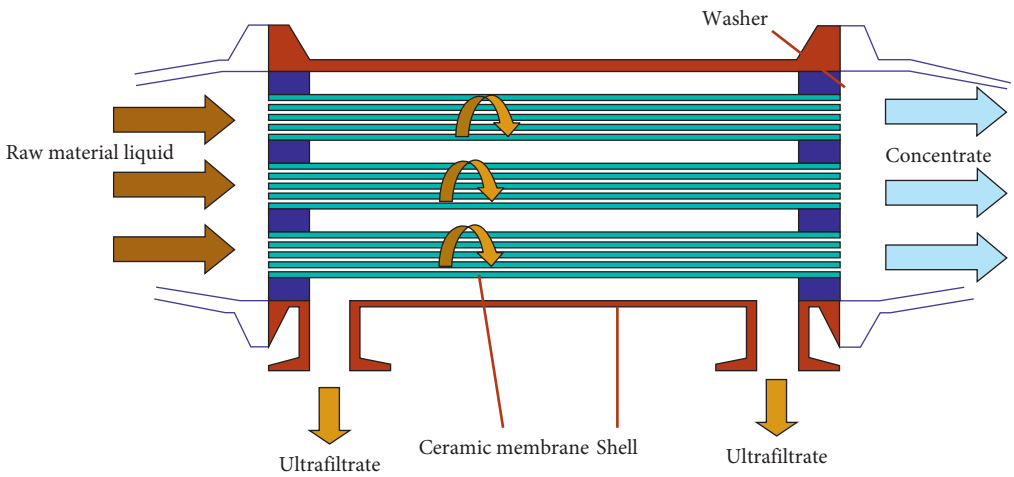


Figure 1: Tubular MBR structure.

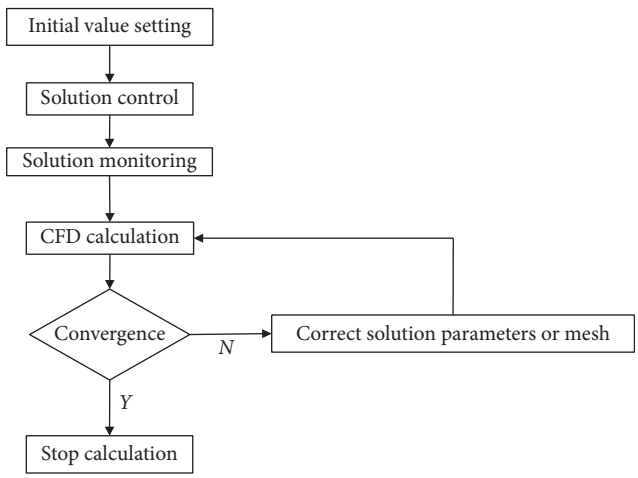


Figure 2: FLUENT solution process block diagram.

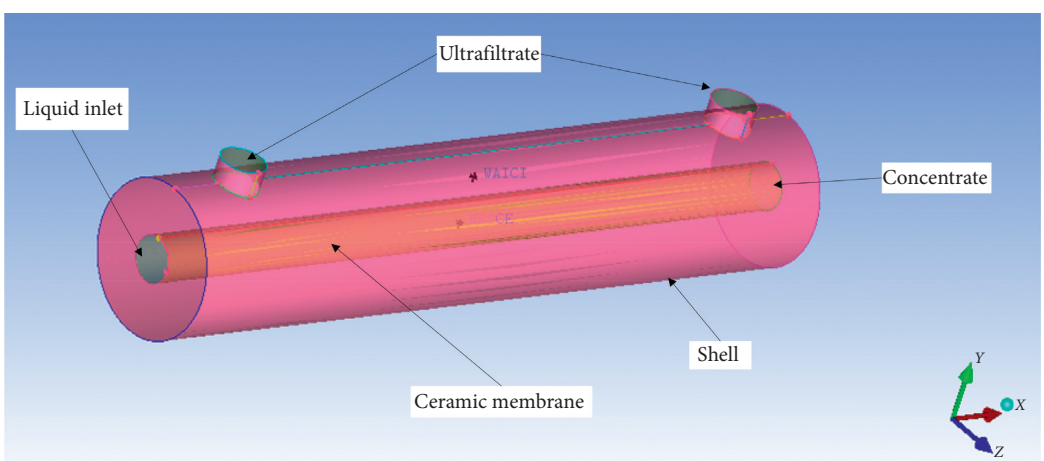


Figure 3: Tubular MBR geometry model.

mode, the SIMPLE algorithm is selected to calculate the mesh file; in the microseparation format option, the gradient is set to the unit-based least squares method. The pressure is set to a second-order format and the momentum equation is set to a second-order upwind style. Secondly, we set the convergence threshold and the monitoring window. Since the research object of this paper is to calculate the water production of the tubular MBR, then this article sets the

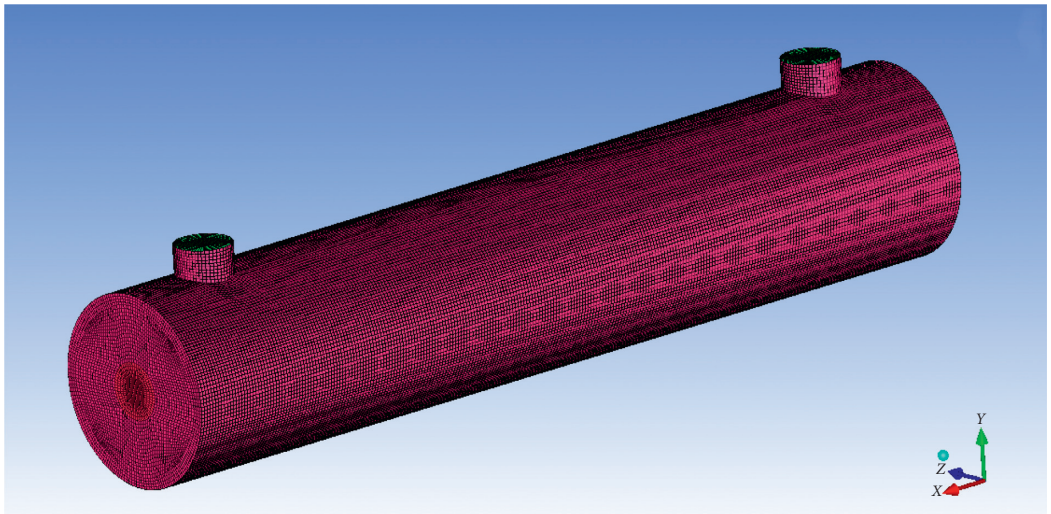


Figure 4: The resulting mesh file.

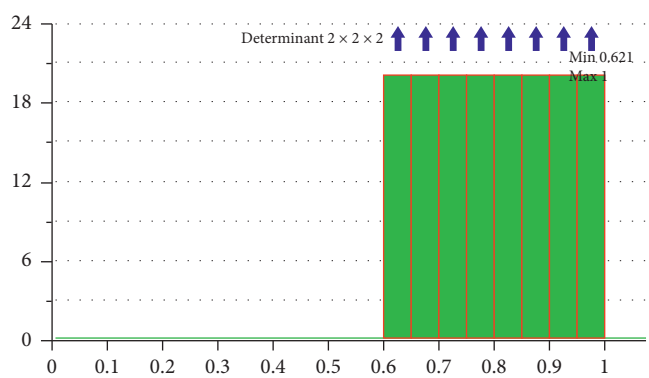


Figure 5: Determinant $2 \times 2 \times 2$ mesh quality map.

traffic monitoring window at the model exit (the mass flow rate is positive for inflow and negative for outflow). And, we set the convergence threshold to 0.001. Then, initialize the flow field and start the iterative calculation. Figure 6 shows the residual curve in the iterative calculation process of the solver. When the residuals of the velocity in the x-axis, y-axis, and z-axis and the continuity of the continuity equation are not greater than the convergence threshold, the solver's calculation reaches convergence. Figure 7 represents the flow at the outlet of the mesh file during the calculation, that is, the water production of the tubular MBR [12]. Finally, the calculation result is saved, and the postprocessing operation is prepared.

4.3. Postprocessor (CFD-Post) Visualization Calculation Results. The postprocessor can present the calculation result in a graphical or image manner. With the postprocessor, the user can effectively observe and analyze the calculation result, thereby more intuitively understanding the calculation result of the solver. Due to the poor performance of the built-in postprocessor in FLUENT, this paper uses CFD-Post for postprocessing operations. In CFD-Post, the FLUENT calculation results are first imported, establishing the flow graph and the pressure cloud map at different sections of the model. Figure 8 is a streamline diagram of water in the model file, where the colored lines on the left represent the magnitude of the velocity, which decreases from top to bottom. And, the flow direction and speed of the water are basically the same as those of the actual tubular MBR system. Figures 9 and 10 are pressure cloud diagrams of the model in which the colored lines on the left side represent the magnitude of the pressure, which decreases sequentially from top to bottom. Observing Figures 9 and 10 which is not difficult to find that since there is a large operating pressure at the entrance of the model, the pressure increases as the process approaches the entrance to the model, which is consistent with the operating principle of the actual tubular MBR. Finally, in order to verify the correctness and reliability of the calculation results, this paper has carried out a lot of experimental analysis.

4.4. Experiment Analysis. In order to verify the correctness and reliability of the FLUENT solution results, this paper selects a large number of actual MBR system operation data

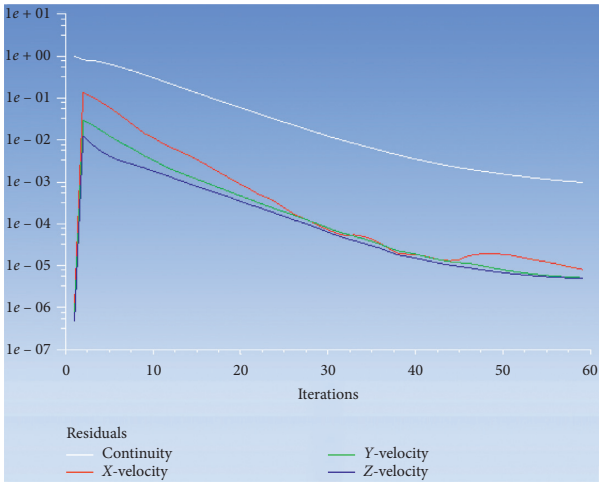


Figure 6: Residual curve.

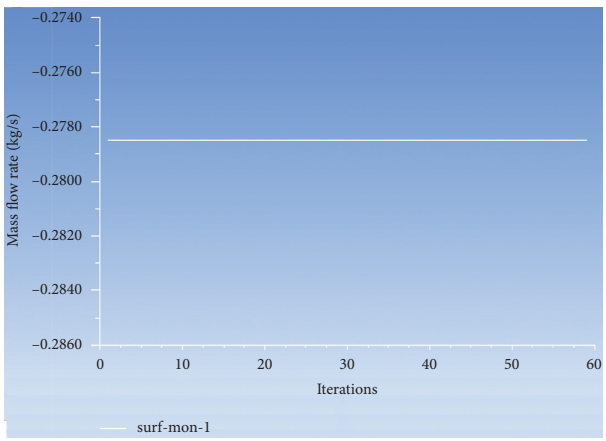


Figure 7: Flow of fluid at the exit of the mesh file.

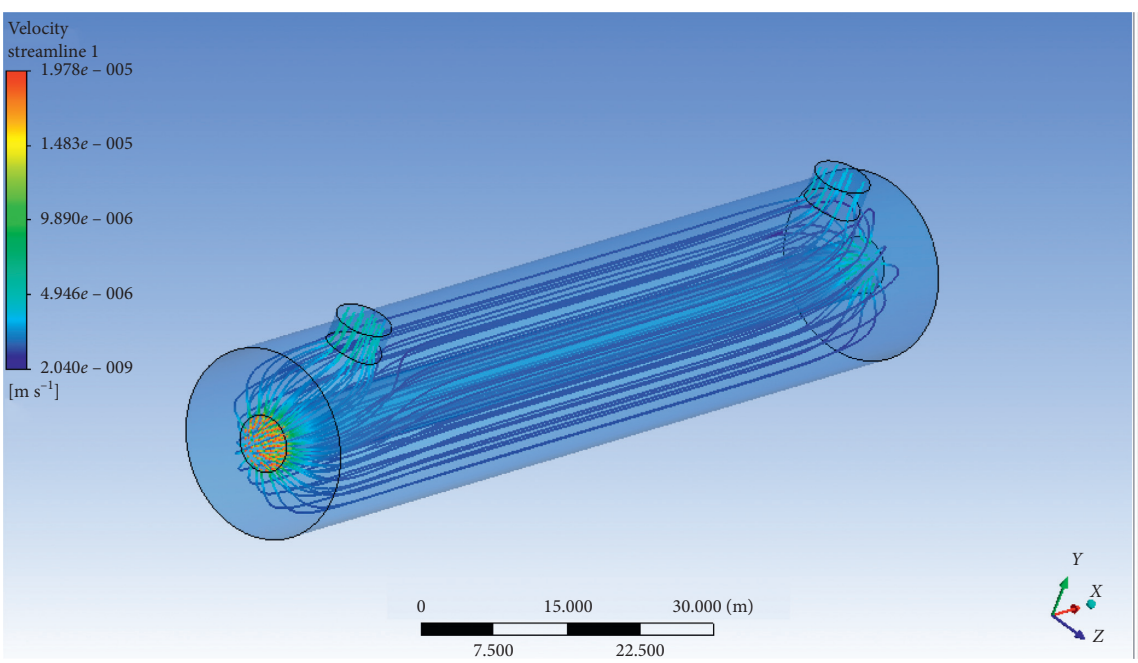


Figure 8: Flow diagram of water in the model.

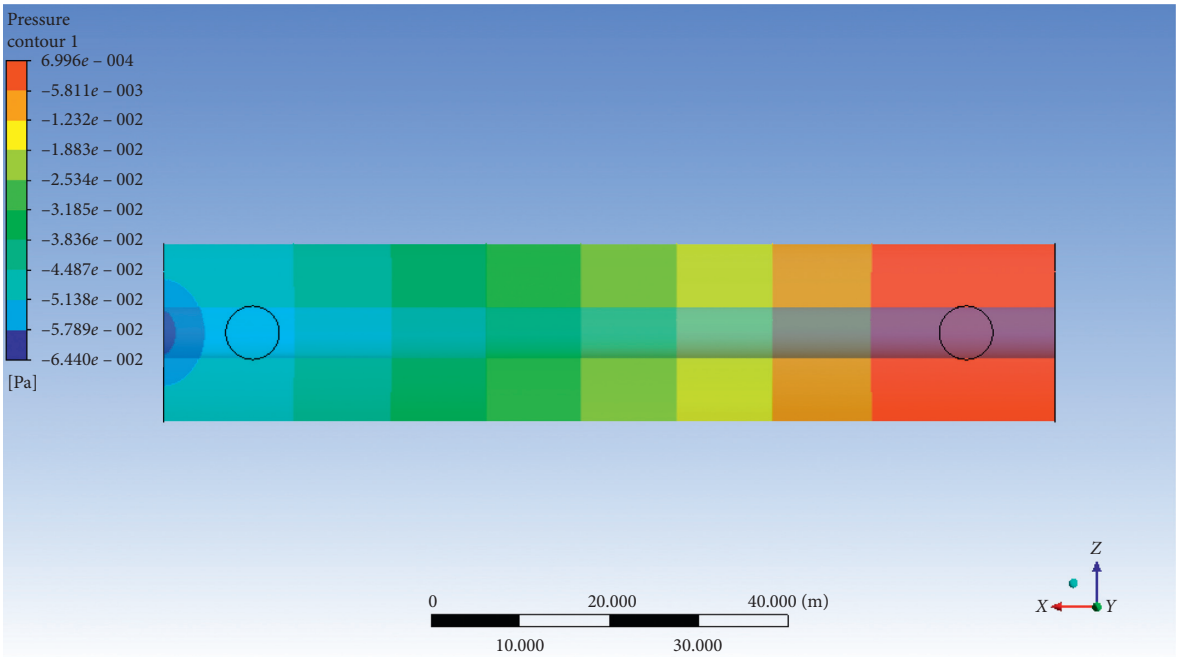


Figure 9: Pressure cloud diagram of the model cross section.

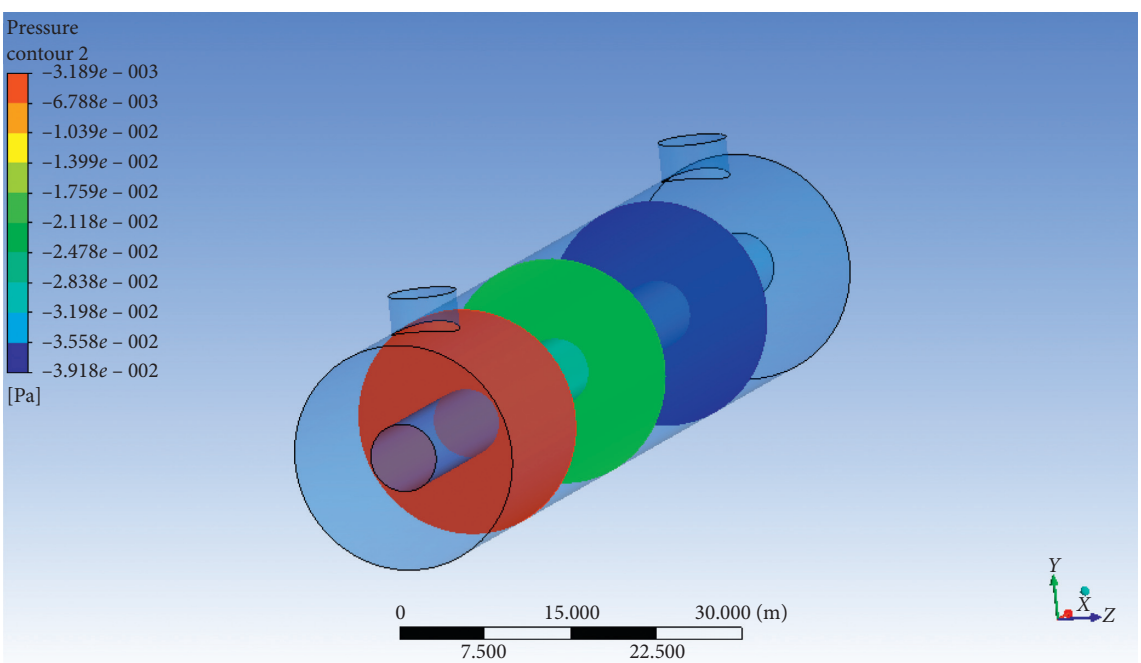


Figure 10: Model different position processing average pressure maps.

of a sewage treatment plant in Shijiazhuang for experimental analysis. In the analysis, the paper divides the actual data, then inputs different water inflows into the inlet boundary conditions of the solver, and starts the iterative calculation. After the calculation, the calculation results obtained by the solver flow monitoring window are compared with the actual tubular MBR system, and the comparison result is shown in Table 1. The comparison results show that the calculation results of the solver are basically the same as those of the actual tubular MBR system, which solves the problem of calculating the water production of tubular MBR model by CFD and achieves the purpose of simulating tubular MBR with CFD, so the calculation result is correct and reliable.

Table 1: Internal pressure series MBR system water production scale.

Water intake (m^3/h)	TMP (Kpa)	Actual water production (m^3/h)	Simulated water production (m^3/h)
2	17	1	1.008
1	17	0.5	0.5004
2.5	17	1.25	1.2492
4	17	2	1.99836
3	17	1.5	1.49508
4.5	17	2.25	2.24964
3.5	17	1.75	1.74816
5	17	2.5	2.50488
7	17	3.5	3.4956
9	17	4.5	4.50036

5. Conclusion

Because the deployment of membrane will affect the water yield of membrane module of tubular MBR, this paper uses the CFD simulation software ANSYS 16.0 to simulate and calculate the tubular MBR. In the process of analysis, firstly, the tube MBR model of a single membrane wire is constructed by ICEM CFD, and the mesh file is obtained by fine mesh division. Then, the fluent solver is used to solve the grid file, and the water yield is obtained. Finally, CFD-post is used to present the calculation results in a graphical way. On this basis, this paper compares the calculation results of the solver with the operation results of the actual MBR system, and the comparison results show that the calculation results of the solver are basically consistent with the operation results of the actual MBR system, realizing the calculation of the water yield of the tubular MBR membrane module, solving the problem of measuring the working efficiency of the tubular MBR system with the water yield, and realizing the work efficiency of the MBR system before deployment. The evaluation has certain reference value for the planning, design, and deployment of MBR. It has certain value and significance for the research of MBR simulation in the future.

Data Availability

The data used to support the findings of this study are included within the supplementary information file.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Feng Kun and Zhang Ming substantially contributed to the conception or design of the work and the acquisition, analysis, or interpretation of data for the work. Li Chunqing and Feng Kun drafted the work or revised it critically for important intellectual content. Liu Xuting approved the final version to be published.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant no. 51378350), the National Natural Science Foundation of China (Grant no. 50808130), and the National Natural Science Foundation of China (Grant no. 21506159).

Supplementary Materials

The supplementary literature is the data of a sewage treatment plant, which is used to set the boundary conditions in fluent to draw the conclusion of the paper. (*Supplementary Materials*)

References

[1] T. Vo, J.-J. Lee, J.-S. Kang, S. Park, and K. Han-Seung, "Nitrogen removal by sulfur-based carriers in a membrane bioreactor (MBR)." *Membranes*, vol. 8, no. 4, 2018.

[2] S. Kitanou, M. Tahri, B Bachiri et al., "Comparative study of membrane bioreactor (MBR) and activated sludge processes in the treatment of Moroccan domestic wastewater," *Water Science and Technology*, vol. 78, no. 5, 2018.

[3] M. Sabaghian, M. R. Mehrnia, M. Esmaieli, and D. Noormohammadi, "Formation and performance of self-forming dynamic membrane (SFDM) in membrane bioreactor (MBR) for treating low-strength wastewater," *Water Science and Technology*, vol. 78, no. 4, 2018.

[4] T. Miura, S. Julien, Le Saux Jean-Claude, M. Le, and S. Le Guyader Françoise, "Virus type-specific removal in a full-scale membrane bioreactor treatment process," *Food And Environmental Virology*, 2017.

[5] F. Yang, Y. Wang, L. Gillerman et al., "Analysis of membrane bioreactor performance for wastewater treatment using ranking methods," *Toxicological & Environmental Chemistry*, vol. 99, no. 7-8, 2017.

[6] D. Gregor, "Wehinger. Radiation matters in fixed-bed CFD simulations," *Chemie Ingenieur Technik*, vol. 91, no. 5, 2019.

[7] T. Raja, "Coupled CFD–FE analysis for the exhaust manifold to reduce stress of a direct injection-diesel engine," *International Journal of Ambient Energy*, vol. 40, no. 4, 2019.

[8] S. Chapela, J. Porteiro, M. Garabatos, D. Patiño, M. A. Gómez, and J. L. Míguez, "CFD study of fouling phenomena in small-scale biomass boilers: experimental validation with two different boilers," *Renewable Energy*, p. 140, 2019.

[9] Z. Miao, S. Kuang, H. Zughbi, and A. Yu, "CFD simulation of dilute-phase pneumatic conveying of powders," *Powder Technology*, p. 349, 2019.

[10] S. Agahzamin and L. Pakzad, "CFD investigation of the gas dispersion and liquid mixing in bubble columns with dense vertical internals," *Chemical Engineering Science*, 2019.

[11] M. Behrang, M. Shirvani, and S. H. Hashemabadi, "Multi-helical-channel dust separator: CFD simulation and experiment," *Chemical Engineering Research and Design*, 2019.

[12] Xi Chu, W. Chen, Y. Shang, J. Hao, and De Zhang, "CFD investigation on reverse flow characteristics in U-tubes under two-phase natural circulation," *Progress in Nuclear Energy*, p. 114, 2019.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 9935271, 7 pages
https://doi.org/10.1155/2021/9935271

Research Article

Adaptive Fuzzy Path Tracking Control for Mobile Robots with Unknown Control Direction

Qifei Du,[1] **Lin Sha**,[2,3] **Wuxi Shi**,[2,3] **and Liankun Sun**[4]

[1]*National Demonstration Center for Experimental Engineering Training Education, TianGong University, Tianjin 300387, China*
[2]*Key Laboratory of Intelligent Control of Electrical Equipent, Tianjin 300387, China*
[3]*School of Control Science and Engineering, Tiangong University, Tianjin 300387, China*
[4]*School of Computer Science &Technology, Tiangong University, Tianjin 300387, China*

Correspondence should be addressed to Lin Sha; shalin@tiangong.edu.cn

Received 10 March 2021; Revised 13 April 2021; Accepted 21 April 2021; Published 29 April 2021

Academic Editor: Guoguang Wen

In order to synthesize controllers for wheeled mobile robots (WMRs), some design techniques are usually based on the assumption that the center of mass is at the center of the robot itself. Nevertheless, the exact position of the center of mass is not easy to measure, thus WMRs is a typical uncertain nonlinear system with unknown control direction. Based on the fast terminal sliding mode control, an adaptive fuzzy path tracking control scheme is proposed for mobile robots with unknown control direction. In this scheme, the fuzzy system is used to approximate unknown functions, and a robust controller is constructed to compensate for the approximation error. The Nussbaum-type functions are integrated into the robust controller to estimate the unknown control direction. It is proved that all the signals in the closed-loop system are bounded, and the tracking error converges to a small neighborhood of the origin in a limited time. The effectiveness of the proposed scheme is illustrated by a simulation example.

1. Introduction

The mobile robot system is a typical nonholonomic system, and the research of tracking control has always remained one of the most challenging tasks in the area of mobile robot system.

According to whether the tracking trajectory is a function of time, tracking control is divided into trajectory tracking and path tracking. For the trajectory tracking, based on the kinematics model, the backstepping method, neural network method, fuzzy neural network method, and input-output linearization method were developed in [1–4]. For the path tracking, when the center of mass of the robot is exactly at the geometric center of the wheel axis, the path tracking problem was studied in [5], while the center of mass of the robot is located on the central axis of the two driving wheels, the path tracking problem was developed in [6]. The assumption in [5, 6] that the center of mass is located on the geometric center of the wheel axis or the central axis of the two driving wheels is a good idea for an actual operating robot system; however, the exact position of the center of mass is not easy to measure when it is actually running; thus, the mobile robot system is a typical uncertain nonlinear system. By using the universal approximation of the fuzzy system [7] and the traditional linear sliding mode control techniques, the tracking control scheme were presented for the mobile robot system with uncertainty [8–12]. However, the traditional linear sliding mode control can only achieve infinite time asymptotic convergence and cannot achieve finite time tracking. Moreover, in order to improve convergence speed, it is necessary to increase the design parameters in the sliding mode control, which can increase the gain of the controller and result in the saturation of the control input. Terminal sliding mode (TSM) [13], as an effective finite-time convergence method, has received wide attention by many scholars [14–25]. In recent years, the fast terminal sliding mode (FTSM) [18] and nonsingular Terminal sliding mode (NTSM) [19] had been applied to robot control. For example, in [19, 21, 22], the tracking control was proposed for manipulator based on NTSM, and a simple

terminal sliding mode control was carried out and was presented for mobile robot in [23–25]. However, the proposed methods in [23–25] were a known system whose center of mass is located at the midpoint of the drive axle. When the adaptive fuzzy system is used to control the robot, the approximation error of the fuzzy system is inevitable. In order to suppress the approximation error, it is usually necessary to introduce a robust controller [26–28]. However, in [26–28], it is necessary to assume that the control direction of the system is known. For a mobile robot system with an uncertain center of mass, the control direction is difficult to obtain. Since Nussbaum first proposed the Nussbaum-type function in 1983 and successfully solved the control of the first-order linear system with unknown control direction [29], the Nussbaum-type function has been widely used in the design of control systems [30–33]. For example, the adaptive fuzzy control was proposed for the multiinput multioutput nonlinear system with unknown control direction [30, 31] and for the strict feedback nonlinear system with unknown control direction [32, 33]. In this paper, the Nussbaum-type function technique is applied to the control of the mobile robot system with uncertain center of mass to solve the control problem of unknown control direction of the mobile robot caused by the uncertain center of mass.

In this paper, the path tracking control was proposed for a mobile robot with uncertain center of mass and unknown control direction. The fuzzy system is used to approximate the unknown function in the robot system, and an indirect adaptive fuzzy controller is designed by using FTSM, designing a robust controller to compensate the fuzzy approximation error, and integrating the Nussbaum-type function into the robust controller to estimate the unknown control direction. Based on the Lyapunov stability analysis, an adaptive law is designed for unknown parameters, and it is proved that this method can not only ensure that all signals in the closed-loop system are bounded but also make the tracking error converge to a small neighborhood of the origin within a finite time. The simulation results verify the effectiveness of the method in this paper.

2. Problem Description

The structure of the wheeled robot is shown in Figure 1. The robot has three wheels, and the front left and right wheels are driving wheels. To provide power for the vehicle body, the rear wheel is a follow-up universal wheel, which can not only move with the movement direction of the front wheel but also ensure the balance of the robot.

Because wheeled robot is affected by loads, its center of mass is usually not at the center of the robot itself. In Figure 1, it is assumed that the center of mass of the robot is at point C, and the midpoint of the connecting axis of the first two driving wheels of the robot is point P. The distance from point P to point C is set as L, and the angle between the robot's forward direction axis and the line PC is set as ϕ. The linear velocity of the progress of robot is u_1, and the angular velocity of the robot rotation is u_2. Let the

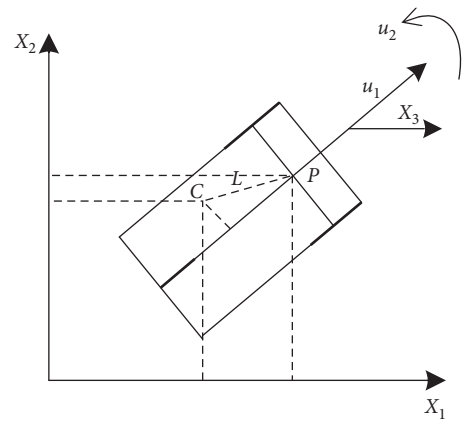


Figure 1: Wheeled mobile robot model.

coordinate of point C as (x_1, x_2, x_3), where x_1 is the abscissa and x_2 is the ordinate, and let the horizontal abscissa be X_1-axis, and the angle between the axis of motion direction of the vehicle body and the X_1-axis is x_3, and the P coordinate is (x_1', x_2', x_3'); then, the kinematics equation of point P is

$$\begin{cases} \dot{x}_1' = u_1 \cos x_3, \\ \dot{x}_2' = u_1 \sin x_3, \\ \dot{x}_3' = u_2. \end{cases} \tag{1}$$

The location relation between point P and C (x_1, x_2, x_3) can be written as

$$\begin{cases} x_1 = x_1' - L\cos x_3 - \phi, \\ x_2 = x_2' - L\sin x_3 - \phi, \\ x_3 = x_3'. \end{cases} \tag{2}$$

By using (1), (2) yields

$$\begin{cases} \dot{x}_1 = u_1 \cos x_3 + Lu_2 \sin(x_3 - \phi), \\ \dot{x}_2 = u_1 \sin x_3 - Lu_2 \cos(x_3 - \phi), \\ \dot{x}_3 = u_2, \end{cases} \tag{3}$$

where $\dot{x}_1$, $\dot{x}_2$, and $\dot{x}_3$ are the velocity components moving in all directions at point C, so (3) is the kinematics model of the robot with the center of mass C (x_1, x_2, x_3) as the reference point.

In order to achieve the ideal control, it is assumed that path function is $f(x_1, x_2) = 0$, and the tracking error function is $z = f(x_1, x_2)$; when the parameters L and ϕ are unknown, under the effect of the amount of control $u(x_1, x_2, x_3)$, the system is made to move along the set path, namely, for a given value φ, there is the time t_1, and when $t > t_1$, the tracking error $z = f(x_1(t), x_2(t)) < \varphi$.

In this paper, it is assumed that the robot runs at linear speed u_1 and the input angular velocity u_2 is controlled. The derivative of the tracking error is

$$\dot{z} = f_{x_1} \cdot \dot{x}_1 + f_{x_2} \cdot \dot{x}_2. \tag{4}$$

Substitute (3) into (4), and we have

$$\dot{z} = \left(f_{x_1}\cos x_3 + f_{x_2}\sin x_3\right)u_1 + \left(f_{x_1}\sin\left(x_3-\phi\right) - f_{x_2}\cos\left(x_3-\phi\right)\right)Lu_2. \tag{5}$$

Let $\mathbf{x} = [x_1, x_2, x_3]^{\mathrm{T}}$ and $h(\mathbf{x}) = (f_{x_1}\cdot\cos x_3 + f_{x_2}\cdot\sin x_3)u_1$:

$$g(\mathbf{x}) = \left(f_{x_1}\cdot\sin\left(x_3-\phi\right) - f_{x_2}\cdot\cos\left(x_3-\phi\right)\right)L. \tag{6}$$

Let $u_2 = u$; then, (5) can be rewritten as

$$\dot{z} = h(\mathbf{x}) + g(\mathbf{x})u. \tag{7}$$

Assumption 1. There is a constant g_0 such that $0 < |g(\mathbf{x})| < g_0$.

The control law is designed as

$$u = \frac{-h(\mathbf{x}) - kz}{g(\mathbf{x})}, \tag{8}$$

where $\mathbf{k} > 0$. Substituting (8) into (7), we obtain $\dot{z} = -\mathbf{k}z$.

Obviously, the tracking error z is going to converge to zero. Since the exact position of the center of mass C is unknown, that is, the distance L and angle ϕ are unknown, the control gain $g(\mathbf{x})$ is actually an unknown nonlinear function, and it difficult to design and implement the control law (8). Considering the universal approximation of the fuzzy system, the fuzzy system is used to approximate the unknown function $g(\mathbf{x})$, and the Nussbaum-type function technique is used to estimate the unknown control direction, and the terminal sliding mode control is used to make the convergence of tracking error in a limited time.

Definition 1 (see [29]). A function $N(\zeta)$ with the following form is called Nussbaum-type function:$\lim_{s\longrightarrow+\infty}\sup(1/s)\int_0^s N(\zeta)\mathrm{d}\zeta = +\infty$ and $\lim_{s\longrightarrow+\infty}\inf(1/s)\int_0^s N(\zeta)\,\mathrm{d}\zeta = -\infty$. The common Nussbaum-type functions are $\zeta^2\cos(\zeta)$, $\zeta^2\sin(\zeta)$, and $\exp(\zeta^2)\cos((\pi/2)\zeta)$.

Lemma 1 (see [34]). *$V(t)$ and $\zeta(t)$ are smooth functions defined in the interval $[0, t_f)$ and $V(t) \geq 0$, $\forall t \in [0, t_f)$, and $N(t) = \exp(\zeta^2)\cos((\pi/2)\zeta)$ is a smooth even Nussbaum-type function. If the following inequality holds, $V(t) \leq c_0 + \int_0^t (g(\tau)N(\zeta(\tau)) + 1)\dot{\zeta}(\tau)d\tau$, $\forall t \in [0, t_f)$, where $g(t)$ is a time-varying parameter, and $g(t) \in I = [l^-, l^+]$, $(0 \notin I)$, c_0 is a reasonable constant; then, $V(t)$, $\zeta(t)$, and $\int_0^t (g(\tau)N(\zeta(\tau)) + 1)\dot{\zeta}(\tau)d\tau$ must be bounded in $[0, t_f)$.*

Lemma 2 (see [20]). *If the continuous function $V(t) > 0$ satisfies the inequality, $\dot{V}(t) + \alpha V(t) + \beta V^{(q/p)}(t) \leq 0$, $\forall t > t_0$, then $V(t)$ will converge to the equilibrium point in finite time t_s, where $t_s \leq t_0 + (p/\alpha(p-q))\ln(\alpha V^{1-(q/p)}(t_0) + \beta/\beta)$, where $\alpha > 0$, $\beta > 0$, p and q are odd, and $q < p$.*

3. Design of Tracking Controller

In order to realize the tracking error convergence in a finite time, the sliding mode surface is designed as

$$s = \dot{z} + \alpha\cdot z + \beta\cdot z^{(q/p)} = 0, \tag{9}$$

where $\alpha > 0$, $\beta > 0$, p and q are odd numbers, and $q < p < 2q$. For the sake of convenience, $\gamma = (q/p)$ are assigned in this paper. The designed control law is designed as

$$u = \frac{1}{g(\mathbf{x})}\left(-h(\mathbf{x}) - \alpha\cdot z - \beta\cdot z^{\gamma}\right). \tag{10}$$

Substituting (10) into (7), we obtain $\dot{z} + \alpha\cdot z + \beta\cdot z^{\gamma} = 0$. According to Lemma 1, z will converge in a limited time.

We know that the nonlinear function $g(\mathbf{x})$ is unknown in (10), so controller (10) cannot be implemented. In this paper, the fuzzy logic system is adopted to approximate the nonlinear function $g(\mathbf{x})$. The form of the fuzzy rule base is as follows: $R^{(l)}$: if x_1 is F_1^l and, ..., and x_n is F_n^l, then y is $G_l, l = 1, 2, 3, \ldots, M$, where $F_i^l (i = 1, 2, \ldots, n)$ and G^l are fuzzy sets, respectively, belonging to functions $\mu_{F_i^l}(x_i)$ and $\mu_{G^l}(y)$, both of which are Gaussian, where M is fuzzy rule number. $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbf{R}^n$ is the input vector of the fuzzy system, and $y \in \mathrm{R}$ is the output variable. Using single-valued fuzzy generator, product inference rule, and central average fuzzy eliminator, the output form of the fuzzy system can be expressed as follows:

$$\widehat{g}(\mathbf{x}|\theta) = \frac{\sum_{l=1}^{M}\widetilde{y}^l\left(\prod_{i=1}^{n}\mu_{F_i^l}(x_i)\right)}{\sum_{l=1}^{M}\left(\prod_{i=1}^{n}\mu_{F_i^l}(x_i)\right)} = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\psi}(\mathbf{x}), \tag{11}$$

where $\theta = [\theta_1, \theta_2, \ldots, \theta_M]^T$ is the adaptive variable vector and $\theta_l = \widetilde{y}^l$ is the point corresponding to the maximum value of μ_{G^l} $\psi(\mathbf{x}) = [\psi_1(\mathbf{x}), \ldots, \psi_M(\mathbf{x})]^T$ is the fuzzy basis function vector, where $\psi_l(\mathbf{x}) = (\prod_{i=1}^{n}\mu_{F_i^l}(x_i)/\sum_{l=1}^{M}\prod_{i=1}^{n}\mu_{F_i^l}(x_i))$, the control law (10) may have singularity problems, so the fuzzy system (11) $\widehat{g}(\mathbf{x}|\theta) = \theta^T\psi(\mathbf{x})$ is adopted to approximate the unknown function $g(\mathbf{x})$, and the equivalent control law is designed as follows:

$$u_{\mathrm{eq}} = \frac{1}{\widehat{g}(\mathbf{x}|\boldsymbol{\theta}) + \varepsilon\cdot\mathrm{sign}(\widehat{g}(\mathbf{x}|\boldsymbol{\theta}))}\left(-h(\mathbf{x}) - \alpha\cdot z - \beta\cdot z^{\gamma}\right), \tag{12}$$

where $\varepsilon > 0$, $\mathrm{sign}(\widehat{g}(\mathbf{x}|\theta)) = \begin{cases} 1 & \widehat{g}(\mathbf{x}|\theta) \geq 0 \\ -1 & \widehat{g}(\mathbf{x}|\theta) < 0 \end{cases}$.

To compensate the approximation error of the fuzzy system, the control law is designed as follows:

$$u = u_{\mathrm{eq}} + u_r, \tag{13}$$

where robust control u_r will be designed later. Substituting (12) into (6), we obtain

$$\dot{z} = -\alpha z - \beta z^{\gamma} - \widehat{g}(\mathbf{x}|\boldsymbol{\theta})u_{\mathrm{eq}} + g(\mathbf{x})u_{\mathrm{eq}} + g(\mathbf{x})u_r - \varepsilon\cdot\mathrm{sign}(\widehat{g}(\mathbf{x}|\boldsymbol{\theta}))u_{\mathrm{eq}}. \tag{14}$$

The optimal parameter of the adaptive vector is defined as $\theta^* = \arg\left\{\min_{\theta\in\Omega_g}[\sup_{x\in\mathbf{D}_x}|\widehat{g}(\mathbf{x}|\theta) - g(\mathbf{x})|]\right\}$, where $\mathbf{D}_x$ is the definition domain of input variables of the fuzzy system and Ω_g is the allowable set of adaptive parameter θ.

Define the minimum approximation error as $\omega = \widehat{g}(\mathbf{x}|\theta^*) - g(\mathbf{x})$, and define $\Phi = \widehat{\theta} - \theta^*$, then (14) can be written as

$$\dot{z} = -\alpha z - \beta z^{\gamma} - \mathbf{\Phi}^{T}\boldsymbol{\psi}(\mathbf{x})u_{\mathrm{eq}} - \omega u_{\mathrm{eq}} - \varepsilon \cdot \mathrm{sign}(\widehat{g}(\mathbf{x}|\theta))u_{\mathrm{eq}} + g(\mathbf{x})u_r. \tag{15}$$

Based on universal approximation theorem, ω is bounded but not easy to measure. In this paper, we assume that there is a constant $\rho^* > 0$ so that $|\omega| \le \rho^*$, due to the unknown ρ^*, define $\widehat{\delta}_g$ is the estimated value of ρ^*, and let $\overline{\delta}_g = \widehat{\delta}_g - \rho^*$.

In (15), the sign of robust control gain $g(\mathbf{x})$ is unknown, which makes it difficult to design a robust controller u_r. However, Nussbaum-type function technique is a feasible method to solve such unknown problems; thus, Nussbaum-type function $N(\zeta) = \exp(\zeta^2)\cos((\pi/2)\zeta)$ is introduced into the design of a robust controller u_r. The robust controller u_r is designed as follows:

$$u_r = N(\zeta)\frac{\left(\widehat{\delta}_g + \varepsilon\right)^2 u_{\mathrm{eq}}^2 z}{\left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| + \sigma^2}, \qquad \dot{\zeta} = \frac{\left(\widehat{\delta}_g + \varepsilon\right)^2 u_{\mathrm{eq}}^2 z^2}{\left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| + \sigma^2}, \tag{16}$$

where σ is the time-varying parameter. The following parameter adaptive law is designed as

$$\dot{\widehat{\theta}} = \eta\boldsymbol{\psi}(\mathbf{x})u_{\mathrm{eq}}z, \tag{17}$$

$$\dot{\widehat{\delta}}_g = \mu\left|u_{\mathrm{eq}}\|z\right|, \tag{18}$$

$$\dot{\sigma} = -\lambda\sigma, \tag{19}$$

where $\eta > 0$, $\mu > 0$, $\lambda > 0$, and $\sigma(0) \ne 0$.

Theorem 1. *The adaptive fuzzy controller (13) and the adaptive law of unknown parameters (17)–(19) are adopted to the robot system (7); then,*

(1) All signals in a closed-loop system are bounded

(2) The tracking error z will converge to a small neighborhood $\{z\|z| \le (w/\alpha - \alpha^*)\}$ of the origin in finite time t_s, where $t_s \le (2/\alpha_0(1-\gamma_0))\ln\ (\alpha_0 V'^{1-\gamma_0}(t_0) + \beta_0/\beta_0)$

Proof. (1) Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}z^2 + \frac{1}{2\eta}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \frac{1}{2\mu}\overline{\delta}_g^2 + \frac{1}{2\lambda}\sigma^2. \tag{20}$$

The derivative of (20) is

$$\dot{V} = z\dot{z} + \frac{1}{\eta}\mathbf{\Phi}^{\mathrm{T}}\dot{\widehat{\theta}} + \frac{1}{\mu}\overline{\delta}_g\dot{\widehat{\delta}}_g + \frac{1}{\lambda}\sigma\dot{\sigma}. \tag{21}$$

From (15), we have

$$z\dot{z} = -\alpha z^2 - \beta z^{\gamma+1} - \mathbf{\Phi}^{\mathrm{T}}\boldsymbol{\psi}(\mathbf{x})u_{\mathrm{eq}}z - \omega u_{\mathrm{eq}}z - \varepsilon \cdot \mathrm{sign}(\widehat{g}(\mathbf{x}|\boldsymbol{\theta}))u_{\mathrm{eq}}z + g(\mathbf{x})u_r z. \tag{22}$$

Substituting (22) into (21), we obtain

$$\dot{V} = -\alpha z^2 - \beta z^{\gamma+1} - \mathbf{\Phi}^{T}\boldsymbol{\psi}(\mathbf{x})u_{\mathrm{eq}}z - \omega u_{\mathrm{eq}}z + g(\mathbf{x})u_r z - \varepsilon \cdot \mathrm{sign}(\widehat{g}(\mathbf{x}|\boldsymbol{\theta}))u_{\mathrm{eq}}z + \frac{1}{\eta}\mathbf{\Phi}^{T}\dot{\widehat{\theta}} + \frac{1}{\mu}\overline{\delta}_g\dot{\widehat{\delta}}_g + \frac{1}{\lambda}\sigma\dot{\sigma}. \tag{23}$$

Substituting (17) into (23) and because $|\omega| \le \rho^*$, we obtain

$$\dot{V} \le -\alpha z^2 - \beta z^{\gamma+1} + (\rho^* + \varepsilon)\left|u_{\mathrm{eq}}\|z\right| + g(\mathbf{x})u_r z + \frac{1}{\mu}\overline{\delta}_g\dot{\widehat{\delta}}_g + \frac{1}{\lambda}\sigma\dot{\sigma}. \tag{24}$$

Since $\overline{\delta}_g = \widehat{\delta}_g - \rho^*$, we obtain

$$\dot{V} \le -\alpha z^2 - \beta z^{\gamma+1} + \left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| - \overline{\delta}_g\left|u_{\mathrm{eq}}\|z\right| + g(\mathbf{x})u_r z + \frac{1}{\mu}\overline{\delta}_g\dot{\widehat{\delta}}_g + \frac{1}{\lambda}\sigma\dot{\sigma}. \tag{25}$$

Substituting (18) into (25), we obtain

$$\dot{V} \le -\alpha z^2 - \beta z^{\gamma+1} + \left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| + g(\mathbf{x})u_r z + \frac{1}{\lambda}\sigma\dot{\sigma}. \tag{26}$$

Substituting the robust control law (16) into (26) yields

$$\dot{V} \le -\alpha z^2 - \beta z^{\gamma+1} + \left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| + \frac{1}{\lambda}\sigma\dot{\sigma} + g(\mathbf{x})N(\zeta)\dot{\zeta}. \tag{27}$$

By using (16), (27) can be rewritten as

$$\dot{V} \le -\alpha z^2 - \beta z^{\gamma+1} + (g(\mathbf{x}(t))N(\zeta) + 1)\dot{\zeta} + \frac{1}{\lambda}\sigma\dot{\sigma} + \left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| - \frac{\left(\widehat{\delta}_g + \varepsilon\right)^2 u_{\mathrm{eq}}^2 z^2}{\left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| + \sigma^2}. \tag{28}$$

Since

$$\left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| - \frac{\left(\widehat{\delta}_g + \varepsilon\right)^2 u_{\mathrm{eq}}^2 z^2}{\left(\widehat{\delta}_g + \varepsilon\right)\left|u_{\mathrm{eq}}\|z\right| + \sigma^2} \le \sigma^2, \tag{29}$$

using (28) and (29), we can obtain

$$\dot{V} \le -\alpha z^2 - \beta z^{\gamma+1} + (g(\mathbf{x}(t))N(\zeta) + 1)\dot{\zeta} + \sigma^2 + \frac{1}{\lambda}\sigma\dot{\sigma}. \tag{30}$$

Substituting (19) into (29), we have

$$\dot{V} \leq -\alpha z^2 - \beta z^{\gamma+1} + (g(\mathbf{x}(t))N(\zeta) + 1)\dot{\zeta}. \quad (31)$$

Integrating (31), we obtain

$$V(t) - V(0) \leq -\int_0^t \alpha z^2 - \int_0^t \beta z^{\gamma+1} + \int_0^t (g(\mathbf{x}(\tau))N(\zeta) + 1)\dot{\zeta}\mathrm{d}\tau. \quad (32)$$

The following resulted from (32):

$$V(t) \leq V(0) - \int_0^t \alpha z^2 - \int_0^t \alpha z^2 - \int_0^t \beta\left(z^{(\gamma+1/2)}\right)^2 + \int_0^t (g(\tau)N(\zeta) + 1)\dot{\zeta}\mathrm{d}\tau \leq V(0) + \int_0^t (g(\tau)N(\zeta) + 1)\dot{\zeta}\mathrm{d}\tau. \quad (33)$$

According to Assumption 1, $g(t) \in [-g_0, g_0]$ and $0 \notin [-g_0, g_0]$; therefore, by using Lemma 1, $V(t)$, $\int_0^t (g(\tau)N(\zeta) + 1)\dot{\zeta}\mathrm{d}\tau$ and $\zeta(t)$ are bounded in the interval $[0, t_f)$; these conclusions are also feasible at $t_f = +\infty$. Therefore, z, $\widehat{\theta}$, $\widehat{\delta}_g$, and σ are also bounded. From (12) and (16), we can know that u_{eq} and u_r are also bounded. Therefore, according to (13), the control law u is bounded.

(2) Since $|\omega| \leq \rho^*$, we get the following equation from (22):

$$z\dot{z} \leq -\alpha z^2 - \beta z^{\gamma+1} + \left\|\mathbf{\Phi}^{\mathrm{T}}\right\|\|\boldsymbol{\psi}(\mathbf{x})\|\left|u_{\mathrm{eq}}\|z\right| + (\rho^* + \varepsilon)\left|u_{\mathrm{eq}}\|z\right| + \left|g(\mathbf{x})\|u_r\|z\right|. \quad (34)$$

(34) can be rewritten as

$$z\dot{z} \leq -z^2\left[\alpha - \left(\left\|\mathbf{\Phi}^{\mathrm{T}}\right\|\|\boldsymbol{\psi}(x)\|\left|u_{\mathrm{eq}}\right| + (\rho^* + \varepsilon)\left|u_{\mathrm{eq}}\right| + \left|g(\mathbf{x})\|u_r\right|\right)|z|^{-1}\right] - \beta z^{\gamma+1}. \quad (35)$$

Let $w = \sup_{t>0}[(\|\Phi^{\mathrm{T}}\|\|\psi(\mathbf{x})\| + \rho^* + \varepsilon)|u_{\mathrm{eq}}| + |g(\mathbf{x})\|u_r|]$; then, (35) can be rewritten as

$$z\dot{z} \leq -\frac{1}{2}z^2 \cdot 2\left(\alpha - w|z|^{-1}\right) - 2^{(\gamma+1/2)}\beta\left(\frac{1}{2}z^2\right)^{(\gamma+1/2)}. \quad (36)$$

If $V' = (1/2)z^2$, from (36), we obtain

$$V' \leq -2\overline{\alpha}(t)V' - 2^{(\gamma+1/2)}\beta(V')^{(\gamma+1/2)}, \quad (37)$$

where $\overline{\alpha}(t) = \alpha - w|z|^{-1}$.

For any small constant $\alpha^* > 0$, an appropriate choice of α is made to $\overline{\alpha}(t) = \alpha - w|z|^{-1} \geq \alpha^* > 0$; thus, it can be obtained from equation (37) that

$$V' \leq -2\alpha^* V' - 2^{(\gamma+1/2)}\beta(V')^{(\gamma+1/2)}. \quad (38)$$

Equation (38) can be written as

$$V' \leq -\alpha_0 V' - \beta_0 (V')^{\gamma_0}, \quad (39)$$

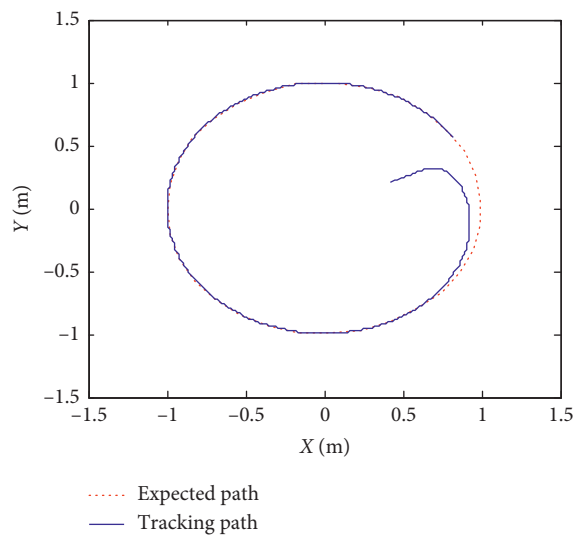


Figure 2: Expected path and tracking path.

where $\alpha_0 = 2\alpha^*$, $\beta_0 = 2^{(\gamma+1/2)}\beta$, and $\gamma_0 = (\gamma + 1/2)$. According to Lemma 2 and $\overline{\alpha}(t) = \alpha - w|z|^{-1} \geq \alpha^* > 0$, a reasonable choice of α is made. Then, the tracking error z will converge to a small neighborhood of the origin $\{z\|z| \leq (w/\alpha - \alpha^*)\}$ in a finite time t_s, where $t_s \leq (2/\alpha_0(1-\gamma_0))\ \ln(\alpha_0 V'^{1-\gamma_0}(t_0) + \beta_0/\beta_0)$.

4. Simulation Experiment

The following simulation experiments will be used to verify the effectiveness of the control method designed in this paper. Let the desired path of the robot be $f(x_1, x_2) = x_1^2 + x_2^2 - 1 = 0$, that is, a circle with radius 1, and set the tracking error as $z = f(x_1, x_2) = x_1^2 + x_2^2 - 1$; then, $\dot{z} = 2u_1(x_1\cos x_3 + x_2\sin x_3) + 2Lu_2[x_1\sin(x_3 - \phi) - x_2\cos(x_3 - \phi)]$. According to the above description, we have $h(\mathbf{x}) = 2u_2(x_1\cos x_3 + x_2\sin x_3)$ and $g(\mathbf{x}) = 2Lu_2[x_1\sin(x_3 - \phi) - x_2\mathrm{co}(x_3 - \phi)]$.

The position of the center of mass C cannot be determined from the previous description, so the control gain $g(\mathbf{x})$ is also unknown. By using the fuzzy system $\widehat{g}(\mathbf{x}|\theta_g) = \theta_g^{\mathrm{T}}\psi(\mathbf{x})$ to approximate $g(\mathbf{x})$, the membership functions inputted are as follows: $\mu_{F_i^1}(x_i) = \exp(-((x_i - 1.5/2))^2)$, $\mu_{F_i^2}(x_i) = \exp(-(x_i/2)^2)$, and $\mu_{F_i^3}(x_i) = \exp(-((x_i + 1.5/2))^2)$, all of which are Gaussian, and $i = 1, 2, 3$, so the fuzzy system designed in this paper has 27 rules.

In the kinematics model (3), it is assumed that $L = 0.3$, $\phi = \pi/6$, the initial attitude coordinate of the robot is $(0.4, 0.2, -(\pi/8))$, and its linear velocity $u_1 = 1$. Set the parameter in the controller $\varepsilon = 0.1$, and each component of the initial value in the parameter $\widehat{\theta}$ is randomly selected within the interval $[-1, 1]$, $\zeta(0) = 1.2$, $\widehat{\delta}_g(0) = 0.1$, and $\sigma(0) = 0.1$ The parameters in the adaptive law are as follows: $\eta = 3.8$, $\mu = 0.6$, and $\lambda = 0.5$.

Under the action of FTSM adaptive fuzzy controller, the robot's tracking effect on the desired path $f(x_1, x_2) =$

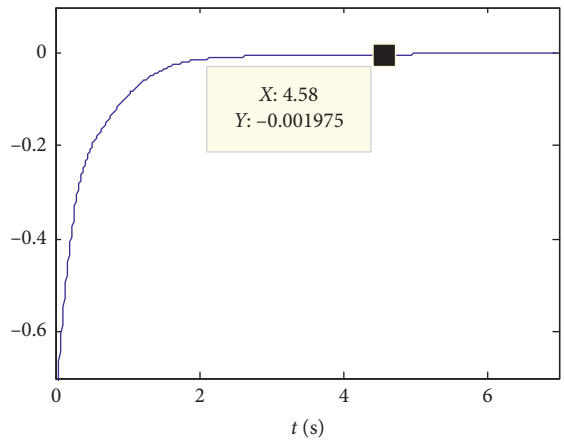


Figure 3: Error curve.

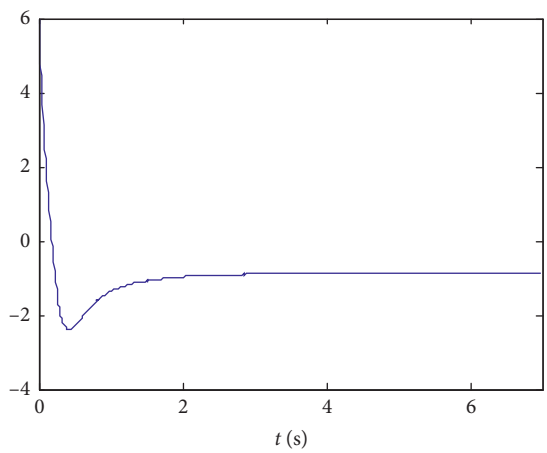


Figure 4: Controller signal.

$x_1^2 + x_2^2 - 1 = 0$, namely, the circular trajectory, which is shown in Figure 2.

Figure 3 shows the error curve of robot path tracking. Figure 4 is the curve of the controller signal u.

From the above simulation results, it can be seen that all the signals in the closed-loop system are bounded, the error convergence of robot path tracking can be achieved, and the tracking error z converges to a small neighborhood of the origin $|z| \leq 1.98 \times 10^{-3}$ after time $t_s = 4.58$ s.

5. Conclusion

In this paper, an adaptive fuzzy path tracking control method has been developed for the mobile robot system with unknown mass center position. In this method, the Nussbaum-type function is used to estimate the unknown control direction, and the terminal sliding mode is used to ensure the convergence of trajectory tracking in a limited time. It is proved that the scheme can not only guarantee that all signals in the closed-loop system are bounded but also make the path tracking error of the mobile robot converge to a small neighborhood near the origin in finite time. The effectiveness of the proposed scheme is illustrated by a simulation. The research of adaptive fuzzy path tracking control method is still in the stage of theoretical research. The next step will be how to introduce engineering practice to solve practical engineering problems.

Data Availability

The data used in this article are simulated, and the authors can provide the source code for the data generation if needed.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (51807138) and Tianjin Nature Science Foundation under Grant 20JCYBJC00180.

References

[1] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomousmobile robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 384–389, Cincinnati, OH, USA, May 1990.

[2] G. F. Yuan, S. X. Yang, and G. S. Mittal, "Tracking control of a mobile robot using a neural dynamics based approach," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pp. 163–168, Seoul, Korea, May 2001.

[3] Y. Hu and S. X. Yang, "A fuzzy neural dynamics based tracking controller for a nonholonomic mobile robot," in *Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 205–210, Kobe, Japan, July 2003.

[4] D.-H. Kim and J.-H. Oh, "Tracking control of a two-wheeled mobile robot using input-output linearization," *Control Engineering Practice*, vol. 7, no. 3, pp. 369–373, 1999.

[5] B. L. Ma and W. Huo, "Path tracking control and stabilization of mobile car," *Robot*, vol. 17, no. 6, pp. 358–362, 1995.

[6] D. Q. Sun, W. Huo, and X. Yang, "Path following control of mobile robots with model uncertainty based on hierarchical fuzzy systems," *Control Theory & Applications*, vol. 21, no. 4, pp. 489–500, 2004.

[7] L. X. Wang, *Adaptive Fuzzy Systems and Control-Design and Stability Analysis*, PTR Prentice-Hall, Hoboken, NJ, USA, 1994.

[8] C.-Y. Chen, T.-H. S. Li, Y.-C. Yeh, and C.-C. Chang, "Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots," *Mechatronics*, vol. 19, no. 2, pp. 156–166, 2009.

[9] N. Yagiz and Y. Hacioglu, "Robust control of a spatial robot using fuzzy sliding modes," *Mathematical and Computer Modelling*, vol. 49, no. 1-2, pp. 114–127, 2009.

[10] M.-K. Chang, "An adaptive self-organizing fuzzy sliding mode controller for a 2-DOF rehabilitation robot actuated by pneumatic muscle actuators," *Control Engineering Practice*, vol. 18, no. 1, pp. 13–22, 2010.

[11] R.-J. Lian, "Design of an enhanced adaptive self-organizing fuzzy sliding-mode controller for robotic systems," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1545–1554, 2012.

[12] J. B. Hu, F. Li, G. L. Wei et al., "Theory and applications of backstepping sliding mode variable structure control for uncertain systems," *Systems Engineering and Electronics*, vol. 36, no. 3, pp. 519–523, 2014.

[13] M. Zak, "Terminal attractors for addressable memory in neural networks," *Physics Letter A*, vol. 133, no. 1-2, pp. 18–22, 1988.

[14] C. W. Tao, J. S. Taur, and M. L. Chan, "Adaptive fuzzy terminal sliding mode controller for linear systems with mismatched time-varying uncertainties," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, no. 1, pp. 1–8, 2004.

[15] S. Yu, X. Yu, B. Shirinzadeh, and Z. Man, "Continuous finite-time control for robotic manipulators with terminal sliding mode," *Automatica*, vol. 41, no. 11, pp. 1957–1964, 2005.

[16] S. H. Yu, S. Liu, and H. Xu, "Adaptive fuzzy trajectory-tracking control of uncertain nonholonomic mobile robots," in *Proceedings of the 2008 6th IEEE International Conference on Industrial Informatics*, pp. 13–16, Daejeon, Korea, July 2008.

[17] V. Nekoukar and A. Erfanian, "Adaptive fuzzy terminal sliding mode control for a class of MIMO uncertain nonlinear systems," *Fuzzy Sets and Systems*, vol. 179, no. 1, pp. 34–49, 2011.

[18] X. H. Yu and Z. H. Man, "Fast termainl sliding-mode control design for nonlinear dynamical system," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 2, pp. 261–265, 2002.

[19] Y. Feng, X. Yu, and Z. Man, "Non-singular terminal sliding mode control of rigid manipulators," *Automatica*, vol. 38, no. 12, pp. 2159–2167, 2002.

[20] S. Yu, X. Yu, and Z. Man, "A fuzzy neural network approximator with fast terminal sliding mode and its applications," *Fuzzy Sets and Systems*, vol. 148, no. 3, pp. 469–486, 2004.

[21] C.-K. Lin, "Nonsingular terminal sliding mode control of robot manipulators using fuzzy wavelet networks," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 849–859, 2006.

[22] M. Jin, J. Lee, P. H. Chang, and C. Choi, "Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3593–3601, 2009.

[23] S. H. Li and Y. P. Tian, "Tracking control of mobile robots," *Control and Decision*, vol. 15, no. 5, pp. 626–628, 2000.

[24] S. H. Li and Y. P. Tian, "Trajectory tracking control of mobile robots in finite time," *Journal of Southeast University*, vol. 34, no. 1, pp. 113–116, 2004.

[25] Q. Y. Wu, M. D. Yan, and Y. Y. He, "Fast terminal sliding mode tracking controller design for nonholonomic mobile robot," *Systems Engineering and Electronics*, vol. 29, no. 12, pp. 2127–2130, 2007.

[26] S. Labiod, M. S. Boucherit, and T. M. Guerra, "Adaptive fuzzy control of a class of MIMO nonlinear systems," *Fuzzy Sets and Systems*, vol. 151, no. 1, pp. 59–77, 2005.

[27] S. Tong, B. Chen, and Y. Wang, "Fuzzy adaptive output feedback control for MIMO nonlinear systems," *Fuzzy Sets Systems*, vol. 156, no. 2, pp. 285–299, 2005.

[28] W. X. Shi, T. Sun, Y. X. Ma et al., "Adaptive fuzzy path following control for a nonholonomic mobile robots," in *Proceedings of the 10th World Congress on Intelligent Control and Automation, WCICA*, pp. 204–208, Beijing, China, July 2012.

[29] R. D. Nussbaum, "Some remarks on a conjecture in parameter adaptive control," *Systems & Control Letters*, vol. 3, no. 5, pp. 243–246, 1983.

[30] A. Boulkroune, M. Tadjine, M. M'Saad et al., "Fuzzy adaptive controller for MIMO nonlinear systems with known and unknown control direction," *Fuzzy Sets Systems*, vol. 161, no. 6, pp. 797–820, 2010.

[31] T.-P. Zhang and Y. Yi, "Adaptive fuzzy control for a class of MIMO nonlinear systems with unknown dead-zones," *Acta Automatica Sinica*, vol. 33, no. 1, pp. 96–99, 2007.

[32] X. Ye and J. Jiang, "Adaptive nonlinear design without a priori knowledge of control directions," *IEEE Transactions on Automatic Control*, vol. 43, no. 11, pp. 1617–1621, 1998.

[33] S. S. Ge, F. Hong, and T. H. Lee, "Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 499–516, 2004.

[34] W. X. Shi, "Observer-based fuzzy adaptive control for multi-input multi-output nonlinear systems with a nonsymmetric control gain matrix and unknown control direction," *Fuzzy Sets and Systems*, vol. 263, pp. 1–26, 2015.