



## Article

# Finite-Time Cluster Synchronization of Fractional-Order Complex-Valued Neural Networks Based on Memristor with Optimized Control Parameters

Qi Chang <sup>1,\*</sup> , Rui Wang <sup>1</sup> and Yongqing Yang <sup>2</sup>

<sup>1</sup> School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China; 2413042015@post.usts.edu.cn

<sup>2</sup> School of Science, Wuxi Engineering Research Center for Biocomputing, Jiangnan University, Wuxi 214122, China; yongqingyang@jiangnan.edu.cn

\* Correspondence: changqi@usts.edu.cn

**Abstract:** The finite-time cluster synchronization (FTCS) of fractional-order complex-valued (FOCV) neural network has attracted wide attention. It is inconvenient and difficult to decompose complex-valued neural networks into real parts and imaginary parts. This paper addresses the FTCS of coupled memristive neural networks (CMNNs), which are FOCV systems with a time delay. A controller is designed with a complex-valued sign function to achieve FTCS using a non-decomposition approach, which eliminates the need to separate the complex-valued system into its real and imaginary components. By applying fractional-order stability theory, some conditions are derived for FTCS based on the proposed controller. The settling time, related to the system's initial values, can be computed using the Mittag-Leffler function. We further investigate the optimization of control parameters by formulating an optimization model, which is solved using particle swarm optimization (PSO) to determine the optimal control parameters. Finally, a numerical example and a comparative experiment are both provided to verify the theoretical results and optimization method.

**Keywords:** fractional-order complex-valued system; coupled memristive neural networks; finite-time cluster synchronization; non-decomposition method; optimization of control parameters



Academic Editor: Aldo Jonathan Muñoz-Vazquez

Received: 28 November 2024

Revised: 5 January 2025

Accepted: 11 January 2025

Published: 14 January 2025

**Citation:** Chang, Q.; Wang, R.; Yang, Y. Finite-Time Cluster Synchronization of Fractional-Order Complex-Valued Neural Networks Based on Memristor with Optimized Control Parameters. *Fractal Fract.* **2025**, *9*, 39. <https://doi.org/10.3390/fractalfract9010039>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Neural networks (NNs) have significant applications across various emerging fields, including multilayer perceptron [1], food quality assessment [2], image recognition [3] and so on [4]. An artificial neural network is a mathematical model that mimics the structure of synaptic connections in the brain to process information. Coupled neural networks (CNNs) enable the interconnection of information between different neurons. With advancements in computers, algorithms, and software, the memristive neural network (MNN) model has gained widespread adoption across diverse domains [5–7]. The existence of memristors was confirmed by HP Labs in 2008, and since then, MNNs have been a focal point of research [8–10]. MNNs offer memory, adaptability, and high parallel processing capabilities. The features of memristors are close to the characteristics of neuronal synapses [11]. Compared to traditional cellular neural networks, MNNs can construct highly integrated nervous systems that more closely approximate the size and structure of the human brain. Fractional calculus extends integer-order calculus, and models described by fractional

differential equations can more accurately simulate real-world systems, particularly viscoelastic models [12]. Complex-valued systems, as an extension of real-valued systems, address certain problems that real-valued systems cannot solve [13,14]. For instance, real-valued systems struggle with symmetry problems and the XOR problem, while a single complex-valued system can handle these issues effectively. In light of these considerations, this paper investigates FOCV-coupled memristive neural networks (FCCMNNs).

Finite-time synchronization (FTS) [15] is notable for its ability to achieve synchronization within a settling time, which has garnered significant attention. Different from asymptotic synchronization, which requires an infinite amount of time, FTS offers practical advantages by reaching synchronization in a predetermined time frame. For effective synchronization within a network, neurons must communicate with one another. In leader-following synchronization scenarios, neurons also need to receive information from a designated leader. Specifically, in the context of cluster synchronization (CS) [16], there may be multiple leaders within each cluster. Consequently, neurons in different clusters must follow the leaders assigned to their respective clusters. Due to its practical implications, CS has attracted considerable interest from researchers [17,18]. This paper addresses the FTCS [19] of FCCMNNs. Neurons within the same cluster can synchronize with their leader's state, achieving synchronization within a settling time that depends on their initial values.

There are various studies that employ the decomposition method, which converts the complex-valued system into two real-valued parts [13,20,21]. However, in practical applications, decomposing a complex-valued system can be challenging. To address this, the Complex-Valued Sign Function (CVSF) [14] has been introduced, allowing for the direct study of complex-valued systems. For example, if  $y = c + di$  is a complex number, the CVSF of  $y$  is  $[y] = [c] + [d]i$ . Additionally, a complex 1-norm [14] is proposed, defined as  $|y|_1 = |c| + |d|i$ . Using the CVSF, a controller is designed to achieve the FTCS of the FCCMNNs without employing the decomposition method. To handle discontinuous terms involving memristors and sign functions, the set-valued map theory [22] and complex-valued Filippov solution [23] are essential. In the study of FTS, it is necessary to calculate the settling time. The Mittag-Leffler function, implemented via the 'mlf' function in MATLAB, provides a means to compute the settling time.

The proposed controller in this paper is a typical negative feedback controller utilizing the CVSF. In the study of network synchronization, many papers design controllers and provide conditions that control parameters must satisfy, often offering a range for these parameters. However, selecting the optimal control parameters within this broad range is a significant challenge that is frequently overlooked. This paper proposes an optimization model to facilitate the selection of control parameters. The fitness function of the model is constructed according to actual control requirements. The constraints of the model are considered by the theoretical results of FTCS. Due to the complexity of the optimization model, an algorithm based on PSO [24] is proposed to find the optimal solution. This method provides guidance for selecting control parameters, leading to an optimal controller that conserves resources. Typically, research on the synchronization of neural networks (NNs) focuses on synchronization criteria, aiming to reduce control energy through various control methods, such as event-triggered control [25], intermittent control [26], impulsive control [27], adaptive control [28], and sampled control [29]. Among these, feedback control [8] is the most commonly used in practical applications. The proposed optimization method offers guidance for parameter selection in feedback control, contributing to more efficient controller design.

It is natural to incorporate time delay into research, given its significant impact on real systems [30]. The study in [31] addressed the FTCS of FOCV NNs without considering

time delay and memristors. Both asymptotic and FTS of coupled NNs, excluding clusters and memristors, were explored in [32], where the focus was on real-valued systems. Additionally, the controller designed in [32] included a time delay, which is not required in this paper. The FTS of FCCMNNs using the decomposition method was investigated in [33]. After decomposition, the research process closely resembles that of real-valued systems. However, there are no prior works that have addressed the FTCS of FCCMNNs using a non-decomposition method. Furthermore, none of the aforementioned studies considered the optimization of control parameters. Therefore, this paper focuses on the FTCS of FCCMNNs, employing a non-decomposition method with optimized control parameters. The major contributions are highlighted below. The difference between this paper and other references is outlined in Table 1.

- ① This paper studies the fractional-order CNNs with a time delay, where the network includes memristive elements and complex-valued states, resulting in a model that is not only more complex than those studied in [31,32] but also better suited for simulating real-world scenarios. Unlike the conventional decomposition approach [20,21,34], this study employs a non-decomposition method. Although the decomposition method simplifies theoretical analysis, it often falls short in practical applications. The non-decomposition approach, on the other hand, provides a more realistic representation of complex systems, aligning more closely with real-world dynamics.
- ② This paper focuses on FTCS, extending beyond conventional CS [35,36] or FTS [37], which has faster convergence time, and the synchronization time can be calculated in advance. To achieve FTCS, we design a simple controller based on the CVSF and derive synchronization criteria that provide a range of control parameters. The settling time for FTCS can be readily determined through the Mittag–Leffler function. While FTCS has been explored in previous studies [19,31,32], the networks investigated in those works differ significantly from the one considered here. Notably, this is the first study to address FTCS for FOCV-coupled neural networks with memristors using a non-decomposition method.
- ③ An optimization model solved by PSO is proposed to select the most cost-effective control parameters. This optimization approach guides the selection of control parameters within a broad range that satisfies the synchronization conditions. Although previous studies [31,32] designed similar controllers, they only provided a range for the parameters without specifically focusing on actual control requirement.

**Table 1.** The difference between this paper and others in the literature.

Item	[19]	[32]	[20]	[31]	[34]	[21]	[37]	This Paper
Fractional-order		✓		✓		✓	✓	✓
Complex-valued			✓	✓	✓	✓	✓	✓
Memristor			✓		✓	✓		✓
FTS	✓	✓	✓		✓		✓	✓
CS	✓	✓		✓				✓
Non-decomposition				✓			✓	✓
Parameter optimization								✓

The remainder of this paper is structured as follows: Section 2 outlines the essential definitions and lemmas, as well as the model of FCCMNNs and the foundational assumptions. Section 3 introduces the designed controller and provides the theorem for achieving FTCS. Section 4 is dedicated to the optimization of control parameters. Section 5 illustrates the effectiveness of the proposed methods through two simulations. Finally, Section 6 provides the conclusions.

**Notations.**  $\mathbb{R}$ ,  $\mathbb{R}^n$ ,  $\mathbb{R}^{n \times n}$  represent the set of real numbers,  $n$ -dimensional real vectors and

$n \times n$  real matrices, respectively.  $N^+$  denotes the set of positive integers.  $\mathbb{C}, \mathbb{C}^n, \mathbb{C}^{n \times n}$  represent the set of complex numbers,  $n$ -dimensional complex vectors and  $n \times n$  complex matrices. If  $y \in \mathbb{C}$ ,  $y = y^R + y^I i$ .  $y^R$  and  $y^I$  means real and imaginary parts of  $y$ ,  $i$  is the imaginary unit.  $|y|_1 = |y^R| + |y^I|$ .  $[y] = [y^R] + [y^I]i$ , and  $[\cdot]$  represents the sign function. If  $Y \in \mathbb{C}^n$ ,  $Y = (y_1, y_2, \dots, y_n)$ .  $Y^H$  is the conjugate transpose of  $Y$ .  $Y^T$  is the transpose of  $Y$ .  $|Y|_1 = \sum_{i=1}^n (|y_i^R| + |y_i^I|)$ .  $[Y] = ([y_1], [y_2], \dots, [y_n])$ .  $\Theta \in \mathbb{R}^{n \times n} > 0$  means that  $\Theta$  is a positive definite matrix.  $\lambda_{\min}(\Theta)$  is the minimum eigenvalue of  $\Theta$ .  $diag(\dots)$  denotes a diagonal matrix. The notation  $\otimes$  is the Kronecker product operator.  $I_N$  is an  $N$ -dimensional identity matrix.  $\mathbf{1}_N$  is an  $N$ -dimensional full one vectors.

## 2. Preliminaries and Model Description

### 2.1. Preliminaries

**Definition 1** ([14]). The sign function of the complex vector  $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$  is defined by  $[x(t)] = ([x_1^R(t)] + [x_1^I(t)]i, [x_2^R(t)] + [x_2^I(t)]i, \dots, [x_n^R(t)] + [x_n^I(t)]i)^T$ , where  $[\cdot] = \text{sign}(\cdot)$  and  $\text{sign}(\cdot)$  is the sign function. Obviously,  $[x^H(t)] = [x(t)]^H$ .

**Definition 2** ([38]). When fractional order  $\alpha \in (0, 1)$ , the Caputo derivative with function  $\omega(s)$  is defined as:

$$D_{t_0,t}^\alpha \omega(s) = \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^t (s-\tau)^{-\alpha} \omega'(\tau) d\tau, \quad (1)$$

where  $t \geq t_0$ , and  $\Gamma(\cdot)$  is the Gamma function that  $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$ .

**Definition 3** ([38]). The two-parameter Mittag–Leffler function is defined by the series expansion

$$E_{p,q}(x) = \sum_{j=0}^{+\infty} \frac{x^j}{\Gamma(pj+q)}, \quad (2)$$

where  $p, q > 0, x \in \mathbb{C}$ . As  $q = 1$ , the one-parameter Mittag–Leffler function is

$$E_p(x) = E_{p,1}(x) = \sum_{j=0}^{+\infty} \frac{x^j}{\Gamma(pj+1)}, \quad (3)$$

where  $\Gamma(\cdot)$  is the same Gamma function with Definition 2.

For any  $x(t) \in \mathbb{C}^n$ , the convex hull of  $[x_j(t)] (j = 1, 2, \dots, n)$  is defined as [14]:

$$\bar{co}([x_j(t)]) = \begin{cases} \{1+i\}, & x_j^R(t) > 0, x_j^I(t) > 0, \\ 1 + \bar{co}\{-1, 1\}i, & x_j^R(t) > 0, x_j^I(t) = 0, \\ \{1-i\}, & x_j^R(t) > 0, x_j^I(t) < 0, \\ \bar{co}\{-1, 1\} + i, & x_j^R(t) = 0, x_j^I(t) > 0, \\ \bar{co}\{-1, 1\} + \bar{co}\{-1, 1\}i, & x_j^R(t) = 0, x_j^I(t) = 0, \\ \bar{co}\{-1, 1\} - i, & x_j^R(t) = 0, x_j^I(t) < 0, \\ \{-1+i\}, & x_j^R(t) < 0, x_j^I(t) > 0, \\ -1 + \bar{co}\{-1, 1\}i, & x_j^R(t) < 0, x_j^I(t) = 0, \\ \{-1-i\}, & x_j^R(t) < 0, x_j^I(t) < 0. \end{cases} \quad (4)$$

**Lemma 1.** For column vector  $\chi(t) : [t_0, +\infty) \rightarrow \mathbb{C}^n$ , there are some properties.

$$\textcircled{1} \quad [31] \quad 2[\chi^H(t)][\chi(t)] = [\chi^H(t)]\omega(t) + \omega^H(t)[\chi(t)], \text{ where } \omega(t) = (\omega_1(t), \omega_2(t), \dots, \omega_n(t))^T \in \overline{c\mathcal{D}}([\chi(t)]).$$

$$\textcircled{2} \quad [14] \quad \chi_p^H(t) + \chi_p(t) = 2\chi_p^R(t) \leq 2|\chi_p(t)|_1, p = 1, 2, \dots, n.$$

$$\textcircled{3} \quad 2|\chi(t)|_1 = [\chi^H(t)]\chi(t) + \chi^H(t)[\chi(t)].$$

$$\textcircled{4} \quad |[\chi(t)]|_1 = [\chi^H(t)][\chi(t)].$$

**Proof.** Based on the definition of  $|\chi(t)|_1$ ,  $[\chi(t)]$  and the Lemma 3 in [14], one has

$\textcircled{3}$

$$2|\chi(t)|_1 = 2 \sum_{p=1}^n |\chi_p(t)|_1 = \sum_{p=1}^n (\overline{[\chi_p(t)]}\chi_p(t) + \overline{\chi_p(t)}[\chi_p(t)]) = [\chi^H(t)]\chi(t) + \chi^H(t)[\chi(t)].$$

$\textcircled{4}$

$$|[\chi(t)]|_1 = \sum_{p=1}^n |[\chi_p(t)]|_1 = \sum_{p=1}^n \overline{[\chi_p(t)]}\chi_p(t) = [\chi^H(t)][\chi(t)].$$

□

**Lemma 2 ([37]).** If there is a continuous and analytic function  $\chi(t) = (\chi_1(t), \chi_2(t), \dots, \chi_n(t))^T \in \mathbb{C}^n$ , then for  $t \in [t_0, +\infty)$ ,  $\alpha \in (0, 1)$ , one has:

$$D_{t_0,t}^\alpha([\chi^H(t)]\chi(t) + \chi^H(t)[\chi(t)]) \leq [\chi^H(t)]D_{t_0,t}^\alpha\chi(t) + D_{t_0,t}^\alpha\chi^H(t)[\chi(t)]. \quad (5)$$

**Lemma 3.** If there is a continuous and differentiable function  $v(t) > 0$ ,  $t \in [t_0, +\infty)$ ,  $v(t_0) > 0$ ,  $\kappa_1, \kappa_2 > 0$ ,  $\alpha \in (0, 1]$ , then for  $D_{t_0,t}^\alpha v(t) \leq -\kappa_1 v(t) - \kappa_2$ , one has

$$\textcircled{1} \quad [32] \quad v(t) \leq \left(v(t_0) + \frac{\kappa_2}{\kappa_1}\right) E_\alpha(-\kappa_1(t-t_0)^\alpha) - \frac{\kappa_2}{\kappa_1}.$$

$$\textcircled{2} \quad \text{There exists } t^* > t_0 \text{ such that } \lim_{t \rightarrow t^*} v(t) \rightarrow 0, \text{ when } t \geq t^*, v(t) \equiv 0. \quad t^* = t_0 + \left(-\frac{\omega}{\kappa_1}\right)^{\frac{1}{\alpha}},$$

where  $\omega = \max \left\{ y \mid E_\alpha(y) = \frac{\kappa_2}{\kappa_1 v(t_0) + \kappa_2} \right\}$ .

**Proof.** As demonstrated in [31], let  $\Sigma(t) = \left(v(t_0) + \frac{\kappa_2}{\kappa_1}\right) E_\alpha(-\kappa_1(t-t_0)^\alpha) - \frac{\kappa_2}{\kappa_1}$ . Since  $\kappa_1 > 0$  and  $\alpha \in (0, 1]$ , it is clear that  $E_\alpha(-\kappa_1(t-t_0)^\alpha)$  is non-increasing. Firstly, the existence of  $t^*$  needs to be verified. Obviously,  $E_\alpha(0) = 1$  and  $\lim_{t \rightarrow +\infty} E_\alpha(-t) = 0$ , then  $\Sigma(t_0) = v(t_0) > 0$ ,  $\lim_{t \rightarrow \infty} \Sigma(t) = -\frac{\kappa_2}{\kappa_1} < 0$ . Consequently, there is  $t^* > t_0$  such that  $\Sigma(t^*) = 0$ . That is to say  $E_\alpha(-\kappa_1(t^*-t_0)^\alpha) = \frac{\kappa_2}{\kappa_1 v(t_0) + \kappa_2}$ . In addition,  $t^* = t_0 + \left(-\frac{\omega}{\kappa_1}\right)^{\frac{1}{\alpha}}$ , where  $\omega = \max \left\{ y \mid E_\alpha(y) = \frac{\kappa_2}{\kappa_1 v(t_0) + \kappa_2} \right\} < 0$ . Secondly, it is proved by contradiction that  $v(t) \equiv 0(t \geq t^*)$ . If  $v(t) \equiv 0(t \geq t^*)$  is not true, then  $\exists t' > t^*$  such that  $v(t') > 0$ . However,  $v(t') \leq \Sigma(t') = \left(v(t_0) + \frac{\kappa_2}{\kappa_1}\right) E_\alpha(-\kappa_1(t'-t_0)^\alpha) - \frac{\kappa_2}{\kappa_1} \leq \left(v(t_0) + \frac{\kappa_2}{\kappa_1}\right) E_\alpha(-\kappa_1(t^*-t_0)^\alpha) - \frac{\kappa_2}{\kappa_1} = 0$ , which is a contradiction with  $v(t') > 0$ . Consequently, there exists  $t^* > t_0$  such that  $\lim_{t \rightarrow t^*} v(t) \rightarrow 0$ , when  $t \geq t^*$ ,  $v(t) \equiv 0$ . □

## 2.2. Model Description

An FCCMNN with a time delay is considered as follows:

$$D_{t_0,t}^\alpha z_i(t) = -Cz_i(t) + A(z_i(t))f(z_i(t)) + B(z_i(t))f(z_i(t-\tau)) + \sigma \sum_{j=1}^N g_{ij}\Gamma z_j(t) + I(t) + u_i(t), \quad (6)$$

where  $\alpha \in (0, 1)$ ,  $i, j = 1, 2, \dots, N$ .  $z_i(t - \tau) = (z_{i1}(t - \tau), z_{i2}(t - \tau), \dots, z_{in}(t - \tau))^T \in \mathbb{C}^n$  and  $z_i(t) = (z_{i1}(t), z_{i2}(t), \dots, z_{in}(t))^T \in \mathbb{C}^n$  are the state of  $i$ th neuron with and without the time delay,  $0 < \tau \leq \tau_{max}$ .  $C = \text{diag}(c_1, \dots, c_n) \in \mathbb{C}^{n \times n}$ .  $f(\cdot) : \mathbb{C}^n \rightarrow \mathbb{C}^n$  represents the activation function.  $\sigma > 0$  is the coupled strength.  $G = (g_{ij})_{N \times N}$  and  $g_{ij} \in \mathbb{R}$ ,  $g_{ij} \neq 0$  means that the neuron  $i$  can receive the information from the neuron  $j$ . Otherwise, there is no link. Define  $g_{ii} = -\sum_{j=1, j \neq i}^N g_{ij}$ , which is equivalent to  $\sum_{j=1}^N g_{ij} = 0$ .  $\Gamma = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_n\} \in \mathbb{R}^{n \times n}$  is an inner matrix and  $\Gamma > 0$ .  $I(t) \in \mathbb{C}^n$  is an external input.  $u_i(t)$  is the controller that needs to be designed.  $A(z_i(t)) = (a_{pq}(z_{ip}(t)))_{n \times n}$ ,  $B(z_i(t)) = (b_{pq}(z_{ip}(t)))_{n \times n}$ , and  $A(z_i(t)), B(z_i(t)) \in \mathbb{C}^{n \times n}$ . Based on the characteristics of memristors,  $a_{pq}(z_{ip}(t))$  and  $b_{pq}(z_{ip}(t))$  are given as follows:

$$a_{pq}(z_{ip}(t)) = \begin{cases} \hat{a}_{pq}, & |z_{ip}(t)|_1 \leq I, \\ \check{a}_{pq}, & |z_{ip}(t)|_1 > I, \end{cases} \quad b_{pq}(z_{ip}(t)) = \begin{cases} \hat{b}_{pq}, & |z_{ip}(t)|_1 \leq I, \\ \check{b}_{pq}, & |z_{ip}(t)|_1 > I, \end{cases} \quad (7)$$

where  $I > 0$  represents the switching jump,  $\hat{a}_{pq}, \check{a}_{pq}, \hat{b}_{pq}$  and  $\check{b}_{pq}$  are all known complex numbers,  $p, q = 1, 2, \dots, n$ . The initial value of network (6) is that  $z_i(w) = \psi_i(w), w \in [t_0 - \tau, t_0], i = 1, 2, \dots, N$ .

The leaders of the network (6) is stated as follows:

$$D_{t_0,t}^\alpha s_l(t) = -Cs_l(t) + A(s_l(t))f(s_l(t)) + B(s_l(t))f(s_l(t-\tau)) + I(t), \quad (8)$$

where  $s_l(t) = (s_{l1}(t), s_{l2}(t), \dots, s_{ln}(t))^T \in \mathbb{C}^n$  is the leader of the  $l$ th cluster. The initial value of network (8) is that  $s_l(w) = \phi_l(w), w \in [t_0 - \tau, t_0], l = 1, 2, \dots, m, m \in \mathbb{N}^+$ . The definitions of  $C, A(\cdot), B(\cdot), f(\cdot), I(t)$  are same with Equation (6).

Suppose that all neurons in network (6) are divided into  $m$  clusters, with neurons within the same cluster following the same leader. For example,

$$\begin{aligned} z_{n_0+1}(t), \dots, z_{n_0+n_1}(t) &\rightsquigarrow s_1(t), \\ z_{n_0+n_1+1}(t), \dots, z_{n_0+n_1+n_2}(t) &\rightsquigarrow s_2(t), \dots, \\ z_{n_0+n_1+\dots+n_{m-1}+1}(t), \dots, z_{n_0+n_1+n_2+\dots+n_m}(t) &\rightsquigarrow s_m(t), \end{aligned} \quad (9)$$

where  $n_l$  is the number of neurons in the  $l$ th cluster,  $l = 1, 2, \dots, m, n_0 = 0, n_0 + n_1 + n_2 + \dots + n_m = N$ . Define  $\Lambda_l = \{n_0 + n_1 + \dots + n_{l-1} + 1, \dots, n_0 + n_1 + \dots + n_l\}$ .

According to the set-valued map theory and complex-valued Filippov solution [34], the networks (6) and (8) can be rewritten to Equations (10) and (11).

$$D_{t_0,t}^\alpha z_i(t) = -Cz_i(t) + \tilde{A}(z_i(t))f(z_i(t)) + \tilde{B}(z_i(t))f(z_i(t-\tau)) + \sigma \sum_{j=1}^N g_{ij}\Gamma z_j(t) + I(t) + u_i(t), \quad (10)$$

$$D_{t_0,t}^\alpha s_l(t) = -Cs_l(t) + \tilde{A}(s_l(t))f(s_l(t)) + \tilde{B}(s_l(t))f(s_l(t-\tau)) + I(t). \quad (11)$$

where  $\tilde{A}(z_i(t)) = (\tilde{a}_{pq}(z_{ip}(t)))_{n \times n}$ ,  $\tilde{B}(z_i(t)) = (\tilde{b}_{pq}(z_{ip}(t)))_{n \times n}$ ,  $\tilde{a}_{pq}(z_{ip}(t)) \in \text{co}[a_{pq}(z_{ip}(t))]$ ,  $\tilde{b}_{pq}(z_{ip}(t)) \in \text{co}[b_{pq}(z_{ip}(t))]$ .  $\text{co}[\cdot]$  is a convex hull. Similarly,  $\tilde{A}(s_l(t)) = (\tilde{a}_{pq}(s_{lp}(t)))_{n \times n}$ ,  $\tilde{B}(s_l(t)) = (\tilde{b}_{pq}(s_{lp}(t)))_{n \times n}$ . It is worth noting that  $\tilde{a}_{pq}(s_{lp}(t)) \neq \tilde{a}_{pq}(z_{ip}(t))$ , and  $\tilde{b}_{pq}(s_{lp}(t)) \neq \tilde{b}_{pq}(z_{ip}(t)), p, q = 1, 2, \dots, n$ .

For convenience, there are some assumptions and propositions.

**Assumption 1.**  $G = (g_{ij})_{N \times N}$  is the adjacent matrix of network (6),

$$G = \begin{pmatrix} G_{11} & G_{12} & \cdots & G_{1m} \\ G_{21} & G_{22} & \cdots & G_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ G_{m1} & G_{m2} & \cdots & G_{mm} \end{pmatrix}, \quad (12)$$

where  $G_{ll} \in \mathbb{R}^{n_l \times n_l}$ ,  $G_{kl} \in \mathbb{R}^{n_k \times n_l}$  ( $k \neq l$ ). Suppose that  $G_{ll}$  and  $G_{kl}$  are both zero row-sum matrices,  $k, l = 1, 2, \dots, m$ . Obviously,  $G$  is a zero row-sum matrix.

**Assumption 2.** ① For  $a, b \in \mathbb{C}$ , there exists  $L > 0$ , such that the activation functions  $f(\cdot)$  satisfies  $|f(a) - f(b)|_1 \leq L|a - b|_1$ . ② There exists  $F > 0$ , such that  $|f(c)|_1 \leq F$  for  $\forall c \in \mathbb{C}$ .

**Proposition 1.** Given Assumptions 1 and 2, the following inequalities can be derived:

①

$$|\widetilde{a}_{pq}(z_{ip}(t))f_q(z_{iq}(t)) - \widetilde{a}_{pq}(s_{lp}(t))f_q(s_{lq}(t))|_1 \leq \bar{a}_{pq}L|e_{iq}(t)|_1 + \check{a}_{pq}F;$$

②

$$|\widetilde{b}_{pq}(z_{ip}(t))f_q(z_{iq}(t - \tau)) - \widetilde{b}_{pq}(s_{lp}(t))f_q(s_{lq}(t - \tau))|_1 \leq 2\bar{b}_{pq}F;$$

where  $\bar{a}_{pq} = \max\{|\hat{a}_{pq}|_1, |\check{a}_{pq}|_1\}$ ,  $\bar{b}_{pq} = \max\{|\hat{b}_{pq}|_1, |\check{b}_{pq}|_1\}$ ,  $\check{a}_{pq} = |\hat{a}_{pq} - \check{a}_{pq}|_1$ ,  $i = 1, 2, \dots, N$ ,  $l = 1, 2, \dots, m$ ,  $p, q = 1, 2, \dots, n$ .

**Proof.**

①

$$\begin{aligned} & |\widetilde{a}_{pq}(z_{ip}(t))f_q(z_{iq}(t)) - \widetilde{a}_{pq}(s_{lp}(t))f_q(s_{lq}(t))|_1 \\ &= |\widetilde{a}_{pq}(z_{ip}(t))f_q(z_{iq}(t)) - \widetilde{a}_{pq}(z_{ip}(t))f_q(s_{lq}(t)) + \widetilde{a}_{pq}(z_{ip}(t))f_q(s_{lq}(t)) - \widetilde{a}_{pq}(s_{lp}(t))f_q(s_{lq}(t))|_1 \\ &= \left| \widetilde{a}_{pq}(z_{ip}(t))(f_q(z_{iq}(t)) - f_q(s_{lq}(t))) + (\widetilde{a}_{pq}(z_{ip}(t)) - \widetilde{a}_{pq}(s_{lp}(t)))f_q(s_{lq}(t)) \right|_1 \\ &\leq \bar{a}_{pq}L|e_{iq}(t)|_1 + \check{a}_{pq}F; \end{aligned}$$

②

$$\begin{aligned} & |\widetilde{b}_{pq}(z_{ip}(t))f_q(z_{iq}(t - \tau)) - \widetilde{b}_{pq}(s_{lp}(t))f_q(s_{lq}(t - \tau))|_1 \\ &\leq |\widetilde{b}_{pq}(z_{ip}(t))|_1 |f_q(z_{iq}(t - \tau))|_1 + |\widetilde{b}_{pq}(s_{lp}(t))|_1 |f_q(s_{lq}(t - \tau))|_1 \\ &\leq 2\bar{b}_{pq}F. \end{aligned}$$

□

Define the synchronization error:  $e_i(t) = z_i(t) - s_l(t)$ ,  $i \in \forall \Lambda_l, l = 1, 2, \dots, m$ . Based on Assumption 1 that  $G$  is a zero row-sum matrix, controlled error network can be described as follows:

$$D_{t_0, t}^\alpha e_i(t) = -Ce_i(t) + \widetilde{A}(e_i(t))\widetilde{f}(e_i(t)) + \widetilde{B}(e_i(t))\widetilde{f}(e_i(t - \tau)) + \sigma \sum_{j=1}^N g_{ij}\Gamma e_j(t) + u_i(t), \quad (13)$$

where  $\widetilde{A}(e_i(t))\widetilde{f}(e_i(t)) = \widetilde{A}(z_i(t))f(z_i(t)) - \widetilde{A}(s_l(t))f(s_l(t))$ ,  $\widetilde{B}(e_i(t))\widetilde{f}(e_i(t - \tau)) = \widetilde{B}(z_i(t))f(z_i(t - \tau)) - \widetilde{B}(s_l(t))f(s_l(t - \tau))$ .

**Definition 4 ([31]).** The networks (6) and (8) achieve the FTCS, if there exists the settling time  $t^* > t_0$  and satisfies:



- ①  $\overline{\lim}_{t \rightarrow \infty} |s_k(t) - s_l(t)|_1 > 0, k \neq l, k, l = 1, 2, \dots, m;$
- ②  $\lim_{t \rightarrow t^*} |x_i(t) - s_l(t)|_1 = 0, i \in \Lambda_l, l = 1, 2, \dots, m;$
- ③  $|x_i(t) - s_l(t)|_1 \equiv 0, t \geq t^*, i \in \Lambda_l, l = 1, 2, \dots, m.$

### 3. Main Results

To achieve the FTCS of the networks (6) and (8), a novel controller with CVSF is designed.

$$u_i(t) = -k_i^1 e_i(t) - k^2 [e_i(t)], \quad (14)$$

where  $k_i^1 > 0, k^2 = \text{diag}(k_1^2, k_2^2, \dots, k_n^2) \in \mathbb{R}^{n \times n} > 0$  is a diagonal matrix. Based on Equation (14), the error network (13) can be written to

$$\begin{aligned} D_{t_0, t}^\alpha e_i(t) = & -C e_i(t) + \tilde{A}(e_i(t)) \tilde{f}(e_i(t)) + \tilde{B}(e_i(t)) \tilde{f}(e_i(t - \tau)) + \sigma \sum_{j=1}^N g_{ij} \Gamma e_j(t) - k_i^1 e_i(t) \\ & - k^2 \mu_i(t), \end{aligned} \quad (15)$$

where  $\mu_i(t) \in \bar{c}o([e_i(t)])$ .

**Theorem 1.** Assuming the validity of Assumptions 1 and 2. Under the designed controller (14), the networks (6) and (8) achieve the FTCS, if

$$\lambda_1 = \lambda_{\min} \left( I_N \otimes (\hat{C} - L\bar{A}) - \sigma \bar{G} \otimes \Gamma + K^1 \otimes I_n \right) > 0, \quad (16)$$

$$\lambda_2 = \lambda_{\min} \left( k^2 - F(\check{A} + 2\bar{B}) \right) > 0. \quad (17)$$

The settling time  $t^* = t_0 + \left(-\frac{\omega}{\lambda_1}\right)^{\frac{1}{\alpha}}$ ,  $\omega = \max \left\{ y \mid E_\alpha(y) = \frac{\lambda_2}{\lambda_1 |e(t_0)|_1 + \lambda_2} \right\}$ , where  $\hat{C} = \text{diag}(c_1^R - |c_1^I|, \dots, c_n^R - |c_n^I|)$ ,  $\bar{A} = (\bar{a}_{pq})_{n \times n}$ ,  $\bar{B} = (\bar{b}_{pq})_{n \times n}$ ,  $\check{A} = (\check{a}_{pq})_{n \times n}$ ,  $\bar{G} = (\bar{g}_{ij})_{N \times N}$ , if  $i = j, \bar{g}_{ii} = g_{ii}$ , otherwise,  $\bar{g}_{ij} = |g_{ij}|$ .  $K^1 = \text{diag}(k_1^1, k_2^1, \dots, k_N^1)$ .

**Proof.** Let Lyapunov function:  $V(t) = 2 \sum_{l=1}^m \sum_{i \in \Lambda_l} |e_i(t)|_1 = \sum_{l=1}^m \sum_{i \in \Lambda_l} ([e_i^H(t)] e_i(t) + e_i^H(t) [e_i(t)])$ . Along with the error network (7) and Lemma 2, it can be concluded that

$$\begin{aligned} D_{t_0, t}^\alpha V(t) \leq & \sum_{l=1}^m \sum_{i \in \Lambda_l} \left\{ [e_i^H(t)] D_{t_0, t}^\alpha e_i(t) + D_{t_0, t}^\alpha e_i^H(t) [e_i(t)] \right\} \\ = & \sum_{l=1}^m \sum_{i \in \Lambda_l} \left\{ - [e_i^H(t)] C e_i(t) - e_i^H(t) C^H [e_i(t)] + [e_i^H(t)] (\tilde{A}(e_i(t)) \tilde{f}(e_i(t))) \right. \\ & + (\tilde{A}(e_i(t)) \tilde{f}(e_i(t)))^H [e_i(t)] + [e_i^H(t)] (\tilde{B}(e_i(t)) \tilde{f}(e_i(t - \tau))) \\ & + (\tilde{B}(e_i(t)) \tilde{f}(e_i(t - \tau)))^H [e_i(t)] + \sigma \sum_{j=1}^N g_{ij} ([e_i^H(t)] \Gamma e_j(t) + e_j^H(t) \Gamma [e_i(t)]) \\ & \left. - k_i^1 ([e_i^H(t)] e_i(t) + e_i^H(t) [e_i(t)]) - ([e_i^H(t)] k^2 \mu_i(t) + \mu_i^H(t) k^2 [e_i(t)]) \right\}. \end{aligned} \quad (18)$$

Calculate each term separately,



$$\begin{aligned}
 & - \sum_{l=1}^m \sum_{i \in \Lambda_l} \left( [e_i^H(t)] C e_i(t) + e_i^H(t) C^H [e_i(t)] \right) \\
 & = - \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n \left( c_q [e_{iq}^H(t)] e_{iq}(t) + c_q^H e_{iq}^H(t) [e_{iq}(t)] \right) \\
 & = 2 \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n \left( -c_q^R |e_{iq}(t)|_1 + c_q^I ([e_{iq}^R(t)] e_{iq}^I(t) - [e_{iq}^I(t)] e_{iq}^R(t)) \right) \\
 & \leq -2 \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n \left( c_q^R |e_{iq}(t)|_1 - |c_q^I| |e_{iq}(t)|_1 \right) = -2 \mathbf{1}_{Nn}^T (I_N \otimes \widehat{C}) \widehat{e}(t),
 \end{aligned} \tag{19}$$

where  $\widehat{e}(t) = (|e_{11}|_1, \dots, |e_{1n}|_1, \dots, |e_{n1}|_1, \dots, |e_{n1n}|_1, \dots, |e_{N1}|_1, \dots, |e_{Nn}|_1)^T \in \mathbb{R}^{Nn}$ .

Based on Lemma 1 and Proposition 1, it is straightforward to derive

$$\begin{aligned}
 & \sum_{l=1}^m \sum_{i \in \Lambda_l} \left( [e_i^H(t)] (\widetilde{A}(e_i(t)) \widetilde{f}(e_i(t))) + (\widetilde{A}(e_i(t)) \widetilde{f}(e_i(t)))^H [e_i(t)] \right) \\
 & = \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{p=1}^n \sum_{q=1}^n \left\{ [e_{iq}^H(t)] (\widetilde{a}_{pq}(z_{ip}(t)) f_q(z_{iq}(t)) - \widetilde{a}_{pq}(s_{lp}(t)) f_q(s_{lq}(t))) \right. \\
 & \quad \left. + (\widetilde{a}_{pq}(z_{ip}(t)) f_q(z_{iq}(t)) - \widetilde{a}_{pq}(s_{lp}(t)) f_q(s_{lq}(t)))^H [e_{iq}(t)] \right\} \\
 & \leq \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{p=1}^n \sum_{q=1}^n \left\{ [e_{iq}^H(t)] (\widetilde{a}_{pq} L |e_{iq}(t)|_1 + \check{a}_{pq} F) + [e_{iq}(t)] (\widetilde{a}_{pq} L |e_{iq}(t)|_1 + \check{a}_{pq} F) \right\} \\
 & = \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{p=1}^n \sum_{q=1}^n \left\{ (\widetilde{a}_{pq} L |e_{iq}(t)|_1 + \check{a}_{pq} F) \times ([e_{iq}^H(t)] + [e_{iq}(t)]) \right\} \\
 & \leq \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{p=1}^n \sum_{q=1}^n \left\{ 2 [e_{iq}(t)]^R (\widetilde{a}_{pq} L |e_{iq}(t)|_1) + 2 [e_{iq}(t)]_1 \check{a}_{pq} F \right\} \\
 & \leq 2L \mathbf{1}_{Nn}^T (I_N \otimes \overline{A}) \widehat{e}(t) + 2F [e^H(t)] (I_N \otimes \check{A}) [e(t)],
 \end{aligned} \tag{20}$$

where  $e(t) = (e_1^T(t), e_2^T(t), \dots, e_N^T(t))^T$ .

Similarly, it can be shown that

$$\begin{aligned}
 & \sum_{l=1}^m \sum_{i \in \Lambda_l} \left( [e_i^H(t)] (\widetilde{B}(e_i(t)) \widetilde{f}(e_i(t - \tau))) + (\widetilde{B}(e_i(t)) \widetilde{f}(e_i(t - \tau)))^H [e_i(t)] \right) \\
 & = \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{p=1}^n \sum_{q=1}^n \left\{ [e_{iq}^H(t)] (\widetilde{b}_{pq}(z_{ip}(t)) f_q(z_{iq}(t - \tau)) - \widetilde{b}_{pq}(s_{lp}(t)) f_q(s_{lq}(t - \tau))) \right. \\
 & \quad \left. + (\widetilde{b}_{pq}(z_{ip}(t)) f_q(z_{iq}(t - \tau)) - \widetilde{b}_{pq}(s_{lp}(t)) f_q(s_{lq}(t - \tau)))^H [e_{iq}(t)] \right\} \\
 & \leq \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{p=1}^n \sum_{q=1}^n \left\{ [e_{iq}^H(t)] (2\overline{b}_{pq} F) + [e_{iq}(t)] (2\overline{b}_{pq} F) \right\} \\
 & \leq \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{p=1}^n \sum_{q=1}^n 4\overline{b}_{pq} F [e_{iq}(t)]_1 = 4F [e^H(t)] (I_N \otimes \overline{B}) [e(t)].
 \end{aligned} \tag{21}$$

According to Lemma 1, it follows that

$$\begin{aligned}
& \sigma \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{j=1}^N g_{ij}([e_i^H(t)]\Gamma e_j(t) + e_j^H(t)\Gamma[e_i(t)]) \\
&= \sigma \sum_{l=1}^m \sum_{i \in \Lambda_l} g_{ii}([e_i^H(t)]\Gamma e_i(t) + e_i^H(t)\Gamma[e_i(t)]) \\
&\quad + \sigma \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{j \in \Lambda_l, i \neq j} g_{ij}([e_i^H(t)]\Gamma e_j(t) + e_j^H(t)\Gamma[e_i(t)]) \\
&\leq 2\sigma \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n g_{ii}\gamma_q |e_{iq}(t)|_1 + 2\sigma \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{j \in \Lambda_l, i \neq j} \sum_{q=1}^n |g_{ij}|\gamma_q |e_{jq}(t)|_1 \\
&= 2\sigma \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n g_{ii}\gamma_q |e_{iq}(t)|_1 + 2\sigma \sum_{l=1}^m \sum_{j \in \Lambda_l} \sum_{i \in \Lambda_l, i \neq j} \sum_{q=1}^n |g_{ji}|\gamma_q |e_{iq}(t)|_1 \\
&= 2\sigma \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{j \in \Lambda_l} \sum_{q=1}^n \bar{g}_{ij}\gamma_q |e_{iq}(t)|_1 = 2\sigma \mathbf{1}_{Nn}^T (\bar{G} \otimes \Gamma) \hat{e}(t).
\end{aligned} \tag{22}$$

Based on Lemma 1, it can also be derived that

$$\begin{aligned}
& - \sum_{l=1}^m \sum_{i \in \Lambda_l} k_i^1 ([e_i^H(t)]e_i(t) + e_i^H(t)[e_i(t)]) \\
&= - \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n k_i^1 ([e_{iq}^H(t)]e_{iq}(t) + e_{iq}^H(t)[e_{iq}(t)]) \\
&= - 2 \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n k_i^1 |e_{iq}(t)|_1 \\
&= - 2 \mathbf{1}_{Nn}^T (K^1 \otimes I_n) \hat{e}(t).
\end{aligned} \tag{23}$$

It is easy to obtain

$$\begin{aligned}
& - \sum_{l=1}^m \sum_{i \in \Lambda_l} ([e_i^H(t)]k^2 \mu_i(t) + \mu_i^H(t)k^2[e_i(t)]) \\
&= - \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n k_q^2 ([e_{iq}^H(t)]\mu_{iq}(t) + \mu_{iq}^H(t)[e_{iq}(t)]) \\
&= - 2 \sum_{l=1}^m \sum_{i \in \Lambda_l} \sum_{q=1}^n k_q^2 |e_{iq}(t)|_1 \\
&= - 2[e^H(t)](I_N \otimes k^2)[e(t)].
\end{aligned} \tag{24}$$

Then, it is straightforward to derive

$$\begin{aligned}
D_{t_0, t}^\alpha V(t) &\leq - 2 \mathbf{1}_{Nn}^T (I_N \otimes (\hat{C} - L\bar{A}) - \sigma(\bar{G} \otimes \Gamma) + K^1 \otimes I_n) \hat{e}(t) \\
&\quad - 2[e^H(t)](I_N \otimes (k^2 - F(\hat{A} + 2\bar{B}))) [e(t)] \\
&\leq - \lambda_1 V(t) - \lambda_2,
\end{aligned} \tag{25}$$

where  $e(t) \in \mathbb{C}^{Nn} \setminus \{0\}$ . Based on Lemma 3, one can obtain

$$|e(t)|_1 \leq \left( |e(t_0)|_1 + \frac{\lambda_2}{\lambda_1} \right) E_\alpha(-\lambda_1(t-t_0)^\alpha) - \frac{\lambda_2}{\lambda_1}. \tag{26}$$

Therefore, the networks (6) and (8) will realize the FTCS with settling time  $t^* = t_0 + \left(-\frac{\omega}{\lambda_1}\right)^{\frac{1}{\alpha}}$ , and  $\omega = \max \left\{ y \left| E_\alpha(y) = \frac{\lambda_2}{\lambda_1 |e(t_0)|_1 + \lambda_2} \right. \right\}$ .  $\square$

**Corollary 1.** When  $\alpha = 1$ , assuming the validity of Assumptions 1 and 2. Under the designed controller (14), the networks (6) and (8) achieve the FTCS, if

$$\lambda_1 = \lambda_{\min} \left( I_N \otimes (\widehat{C} - L\bar{A}) - \sigma \bar{G} \otimes \Gamma + K^1 \otimes I_n \right) > 0, \quad (27)$$

$$\lambda_2 = \lambda_{\min} \left( k^2 - F(\check{A} + 2\bar{B}) \right) > 0. \quad (28)$$

The settling time  $t^* = t_0 + \frac{1}{\lambda_1} \ln \left( \frac{\lambda_1 |e(t_0)|_1 + \lambda_2}{\lambda_2} \right)$ .

**Remark 1.** When  $\alpha = 1$ , the FCCMNN (6) degenerates into a first-order complex-valued neural network model. Thus, the FTCS of an integer-order neural network can be regarded as a special case of Theorem 1.

If networks (6) and (8) exclude the time delay  $\tau$ , they should be modified to

$$D_{t_0,t}^\alpha z_i(t) = -Cz_i(t) + A(z_i(t))f(z_i(t)) + B(z_i(t))f(z_i(t)) + \sigma \sum_{j=1}^N g_{ij} \Gamma z_j(t) + I(t) + u_i(t), \quad (29)$$

$$D_{t_0,t}^\alpha s_l(t) = -Cs_l(t) + A(s_l(t))f(s_l(t)) + B(s_l(t))f(s_l(t)) + I(t). \quad (30)$$

Then, the error network becomes

$$D_{t_0,t}^\alpha e_i(t) = -Ce_i(t) + \tilde{A}(e_i(t))\tilde{f}(e_i(t)) + \tilde{B}(e_i(t))\tilde{f}(e_i(t)) + \sigma \sum_{j=1}^N g_{ij} \Gamma e_j(t) + u_i(t), \quad (31)$$

where  $\tilde{B}(e_i(t))\tilde{f}(e_i(t)) = \tilde{B}(z_i(t))f(z_i(t)) - \tilde{B}(s_l(t))f(s_l(t))$ . Other parameters are same with Equations (6), (8), and (13).

**Corollary 2.** Assuming the validity of Assumptions 1 and 2. Under the designed controller (14), the networks (29) and (30) achieve the FTCS, if

$$\lambda_1 = \lambda_{\min} \left( I_N \otimes (\widehat{C} - L\bar{A} - L\bar{B}) - \sigma \bar{G} \otimes \Gamma + K^1 \otimes I_n \right) > 0, \quad (32)$$

$$\lambda_2 = \lambda_{\min} \left( k^2 - F(\check{A} + \check{B}) \right) > 0. \quad (33)$$

The settling time  $t^* = t_0 + \left(-\frac{\omega}{\lambda_1}\right)^{\frac{1}{\alpha}}$ ,  $\omega = \max \left\{ x \left| E_\alpha(x) = \frac{\lambda_2}{\lambda_1 |e(t_0)|_1 + \lambda_2} \right. \right\}$ , where  $\check{B} = (\check{b}_{pq})_{n \times n}$ ,  $\check{b}_{pq} = |\check{b}_{pq} - \bar{b}_{pq}|_1$ .

**Remark 2.** Corollary 2 encompasses Theorem 1 in [31] as a particular case since the memristive items are not considered in [31].

If there is no memristor in the coupled networks (6) and (8), then the networks should be changed to

$$D_{t_0,t}^\alpha z_i(t) = -Cz_i(t) + Af(z_i(t)) + Bf(z_i(t - \tau)) + \sigma \sum_{j=1}^N g_{ij} \Gamma z_j(t) + I(t) + u_i(t), \quad (34)$$

$$D_{t_0,t}^\alpha s_I(t) = -Cs_I(t) + Af(s_I(t)) + Bf(s_I(t - \tau)) + I(t). \quad (35)$$

Then, the error network is

$$D_{t_0,t}^\alpha e_i(t) = -Ce_i(t) + A\tilde{f}(e_i(t)) + B\tilde{f}(e_i(t - \tau)) + \sigma \sum_{j=1}^N g_{ij}\Gamma e_j(t) + u_i(t), \quad (36)$$

where  $A, B \in \mathbb{C}^{n \times n}$ ,  $\tilde{f}(e_i(t)) = f(z_i(t)) - f(s_I(t))$ ,  $\tilde{f}(e_i(t - \tau)) = f(z_i(t)) - f(s_I(t - \tau))$ . Other parameters are same with Equations (6), (8), and (13).

**Corollary 3.** Assuming the validity of Assumptions 1 and 2. Under the designed controller (14), the networks (34) and (35) achieve the FTCS, if

$$\lambda_1 = \lambda_{\min} \left( I_N \otimes (\hat{C} - LA) - \sigma \bar{G} \otimes \Gamma + K^1 \otimes I_n \right) > 0, \quad (37)$$

$$\lambda_2 = \lambda_{\min} \left( k^2 - 2FB \right) > 0. \quad (38)$$

The settling time  $t^* = t_0 + \left( -\frac{\omega}{\lambda_1} \right)^{\frac{1}{\alpha}}$ , where  $\omega = \max \left\{ x \mid E_\alpha(x) = \frac{\lambda_2}{\lambda_1 |e(t_0)|_1 + \lambda_2} \right\}$ .

**Remark 3.** Corollary 3 is similar to Theorem 2 in [32], with the key difference being that the controller in [32] included a time delay, which can be challenging to implement in practice. The three corollaries demonstrate that Theorem 1 proposed in this paper is more comprehensive and capable of addressing more complex problems.

## 4. Optimization of Control Parameters

In the previous section, the FTCS conditions for networks (6) and (8) were established. Then, one can choose control parameters based on Equations (16) and (17) in Theorem 1. However, these equations only specify the range of the control parameters, which could lead to inefficient use of control resources if the parameters are chosen arbitrarily. To address this issue, an optimization model is developed and solved by PSO.

### 4.1. The Optimization Model

The optimization model of control parameters is structured as follows:

$$\min J = \frac{1}{2} \int_{t_0}^{t^*} (e^H(t)e(t) + u^H(t)u(t))dt, \quad (39)$$

$$\text{s.t. } \lambda_1 > 0, \quad (40)$$

$$\lambda_2 > 0, \quad (41)$$

$$k_i^1 \geq 0, i = 1, 2, \dots, N, \quad (42)$$

$$k_j^2 \geq 0, j = 1, 2, \dots, n, \quad (43)$$

$$t^* = t_0 + \left( -\frac{\omega}{\lambda_1} \right)^{\frac{1}{\alpha}}, \quad (44)$$

$$\omega = \max \left\{ y \mid E_\alpha(y) = \frac{\lambda_2}{\lambda_1 |e(t_0)|_1 + \lambda_2} \right\}, \quad (45)$$

where  $u(t) = (u_1^T(t), u_2^T(t), \dots, u_N^T(t))^T$ ,  $e(t), \lambda_1, \lambda_2, k_i^1, k_j^2$  are same with Theorem 1. Equation (39) is the optimized target function [15], in which  $\frac{1}{2} \int_{t_0}^{t^*} (e^H(t)e(t))dt$  is the integral square error index, and  $\frac{1}{2} \int_{t_0}^{t^*} (u^H(t)u(t))dt$  represents the control energy. The target is to minimize the sum of these two terms. Equations (40)–(45) serve as the constraint

conditions, with Equations (40) and (41) corresponding to the FTCS conditions outlined in Theorem 1. An algorithm based on the PSO method is designed to solve this complicated optimization model.

#### 4.2. An Algorithm with PSO

The PSO algorithm is a classical intelligent optimization technique that leverages the collective intelligence of a group to find the optimal solution. In this algorithm, a large number of particles are iteratively updated. Each particle has its own velocity and position, where the position represents the value of a potential solution, and the velocity updates the position in each iteration. The velocity update is guided by the particle's historical best position  $\mathbf{P}_{best}$  and the global best position  $\mathbf{G}_{best}$ . The following outlines the rules for updating position and velocity:

$$\begin{aligned}\mathcal{X}_i(t+1) &= \mathcal{X}_i(t) + \mathcal{V}_i(t+1), \\ \mathcal{V}_i(t+1) &= w(t)\mathcal{V}_i(t) + c_1rand_1(\mathbf{P}_{besti}(t) - \mathcal{X}_i(t)) + c_2rand_2(\mathbf{G}_{bestj}(t) - \mathcal{X}_i(t)),\end{aligned}\quad (46)$$

where  $\mathcal{V}_i(t) = (\mathcal{V}_{i1}(t), \mathcal{V}_{i2}(t), \dots, \mathcal{V}_{i(N+n)}(t))^T \in \mathbb{R}^{N+n}$  is the velocity vector of the  $i$ th particle,  $\mathcal{X}_i(t) = (\mathcal{X}_{i1}(t), \mathcal{X}_{i2}(t), \dots, \mathcal{X}_{i(N+n)}(t))^T \in \mathbb{R}^{N+n}$  is the position vector of the  $i$ th particle at the  $t$ th iteration.  $N+n$  represents the size of the solution space.  $i = 1, 2, \dots, \mathcal{M}$ ,  $\mathcal{M}$  denotes particle size.  $w(t)$  represents the inertia weight, which is randomly chosen from  $(0.4, 0.9)$ .  $c_1, c_2, rand_1$  and  $rand_2$  are constants.  $\mathbf{P}_{besti}(t) = (p_{besti1}, p_{besti2}, \dots, p_{besti(N+n)})^T$  is the historical optimal position vector of  $i$ th particle.  $\mathbf{G}_{bestj}(t) = (g_{bestj1}, g_{bestj2}, \dots, g_{bestj(N+n)})^T$  is the global optimal position vector of the  $j$ th iteration,  $j = 1, 2, \dots, N_m, N_m$  is the maximum iterations.

To address the optimization model presented in Section 4.1, a novel algorithm is developed based on PSO. The flow chart of the algorithm is outlined in Figure 1.

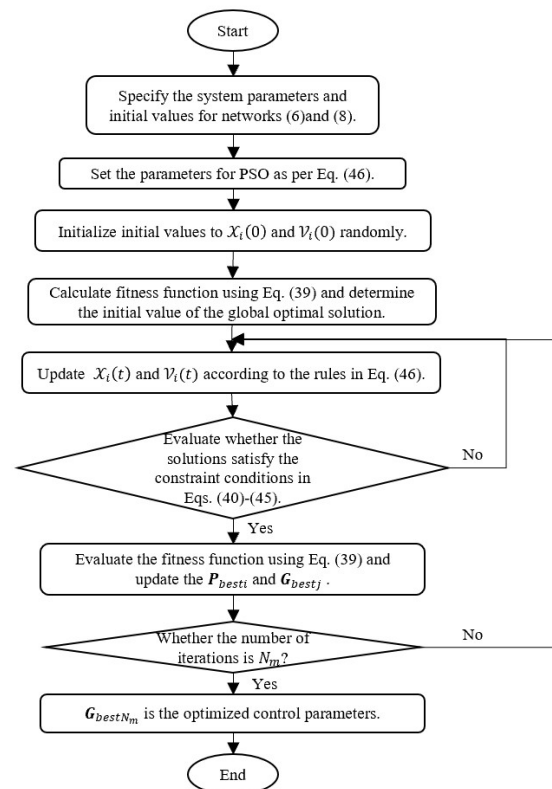


Figure 1. The flow chart of control parameters selection algorithm based on PSO.

**Remark 4.** There is a negative feedback control with CVSF in [31,32]. However, these papers only provide a range for the control parameters, such as  $k_i^1, k_i^2 \geq X$ , where  $X$  is a non-negative constant, leaving the selection of specific control parameters within this range up to the user. The method outlined in this paper facilitates the calculation of control parameters aimed at minimizing the combined costs of control energy and dynamic errors, providing a more economical strategy for selecting control parameters in FCCMNN. This algorithm is based on PSO algorithm and combined the Theorem 1's conditions. It is not a global optimal algorithm and still has some limitations, but it can give a method to select the optimized control parameters.

## 5. Simulation

### 5.1. Example 1

There is an example to demonstrate the feasibility of the main result and the optimization method. The leader's dynamical network (8) is that:

$$D_{t_0,t}^\alpha s_l(t) = -Cs_l(t) + A(s_l(t))f(s_l(t)) + B(s_l(t))f(s_l(t - \tau)) + I(t). \quad (47)$$

Let fractional-order  $\alpha = 0.97$ , the dimension of neuronal states  $n = 2$ , the number of clusters  $m = 2$ . The initial values are  $s_1 = (-1.05 - \mathbf{i}, 0.8 - 2\mathbf{i})^T$ ,  $s_2 = (2.74 + 1.05\mathbf{i}, -1.75 + 1.31\mathbf{i})^T$ .  $C = \text{diag}(0.4 - 3.07\mathbf{i}, 0.2 - 1.43\mathbf{i})$ ,  $f(s(t)) = \tanh(s^R(t)) + \tanh(s^I(t))\mathbf{i}$ , then  $L = 1$  and  $F = 2$ . Let  $\tau = 0.1$ ,  $I(t) = 0.1 \sin(t) + 0.3 \cos(t)\mathbf{i}$ .

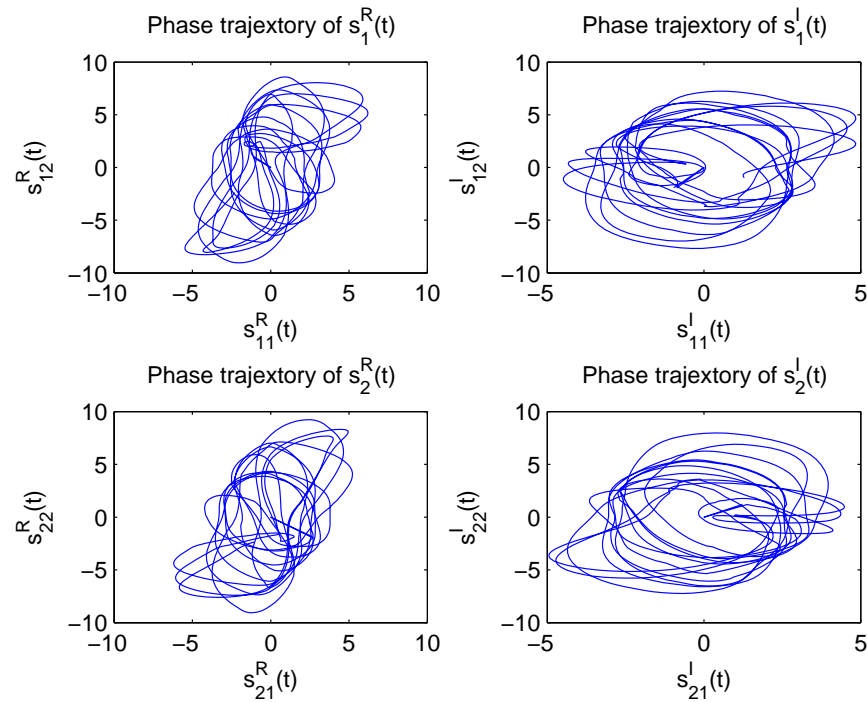
$$\begin{aligned} a_{11}(y_1) &= \begin{cases} -1.24 - 1.75\mathbf{i}, & |y_1|_1 \leq 3, \\ -1.3 - 1.13\mathbf{i}, & |y_1|_1 > 3, \end{cases} & a_{12}(y_1) &= \begin{cases} -2.17 - 1.89\mathbf{i}, & |y_1|_1 \leq 3, \\ -2.22 - 1.99\mathbf{i}, & |y_1|_1 > 3, \end{cases} \\ a_{21}(y_2) &= \begin{cases} -1.56 + 1.69\mathbf{i}, & |y_2|_1 \leq 3, \\ -1.62 + 1.87\mathbf{i}, & |y_2|_1 > 3, \end{cases} & a_{22}(y_2) &= \begin{cases} 1.43 - 0.44\mathbf{i}, & |y_2|_1 \leq 3, \\ 1.3 - 0.06\mathbf{i}, & |y_2|_1 > 3, \end{cases} \\ b_{11}(y_1) &= \begin{cases} 0.06 + 0.31\mathbf{i}, & |y_1|_1 \leq 3, \\ -0.03 + 0.25\mathbf{i}, & |y_1|_1 > 3, \end{cases} & b_{12}(y_1) &= \begin{cases} -0.31 + 0.09\mathbf{i}, & |y_1|_1 \leq 3, \\ 0.22 - 0.07\mathbf{i}, & |y_1|_1 > 3, \end{cases} \\ b_{21}(y_2) &= \begin{cases} 0.03 + 0.44\mathbf{i}, & |y_2|_1 \leq 3, \\ 0.09 + 0.07\mathbf{i}, & |y_2|_1 > 3, \end{cases} & b_{22}(y_2) &= \begin{cases} -0.1 + 0.08\mathbf{i}, & |y_2|_1 \leq 3, \\ -2.21 - 0.07\mathbf{i}, & |y_2|_1 > 3. \end{cases} \end{aligned}$$

The two-dimensional (2D) state curves of  $s(t)$  are shown in Figure 2. The network (6) with the controller (14) is that:

$$\begin{aligned} D_{t_0,t}^\alpha z_i(t) &= -Cz_i(t) + A(z_i(t))f(z_i(t)) + B(z_i(t))f(z_i(t - \tau)) + \sigma \sum_{j=1}^N g_{ij}\Gamma z_j(t) + I(t) \\ &\quad - k_i^1 e_i(t) - k_i^2 [e_i(t)], \end{aligned} \quad (48)$$

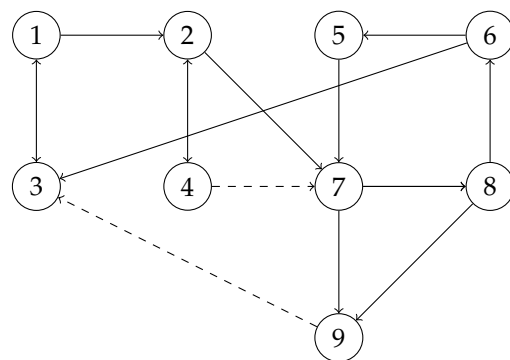
where the strength of coupling  $\sigma = 1.5$ ,  $\Gamma = \text{diag}(0.2, 0.2)$ . The topology of the network (6) is in Figure 3. The solid line represents a connection weight of 1, while the dashed line denotes  $-1$ . The nine neurons are divided into two clusters.  $\Lambda_1 = \{1, 2, 3, 4\}$ ,  $\Lambda_2 = \{5, 6, 7, 8, 9\}$ . By calculation, one can have  $\hat{C} = \text{diag}(-2.67, -1.23)$ . Select  $K^1 = \text{diag}(9.87, 10.17, 10.47, 9.87, 9.87, 9.87, 10.47, 9.87, 10.17)$  and  $K^2 = \text{diag}(4.84, 4.6)$ . Then,  $\lambda_1 = 1.4807 > 0$ ,  $\lambda_2 = 0.1 > 0$ . Let  $t_0 = 0$ , the initial value of  $z_{ip}$  ( $i = 1, 2, \dots, 9, p = 1, 2$ ) is selected randomly from  $[-2, 2] + [-2, 2]\mathbf{i}$ . Based on Theorem 1, one can calculate that  $|e(t_0)|_1 = 67.5107$ ,  $\omega = -32.47$ ,  $t^* = 24.1267$ .

$$\bar{A} = \begin{pmatrix} 2.99 & 4.21 \\ 3.49 & 1.87 \end{pmatrix}, \bar{B} = \begin{pmatrix} 0.37 & 0.4 \\ 0.47 & 0.28 \end{pmatrix}, \tilde{A} = \begin{pmatrix} 0.68 & 0.15 \\ 0.24 & 0.51 \end{pmatrix}.$$



**Figure 2.** The 2D state curves of the leaders  $s_1(t)$  and  $s_2(t)$  of 2 clusters in Example 1.

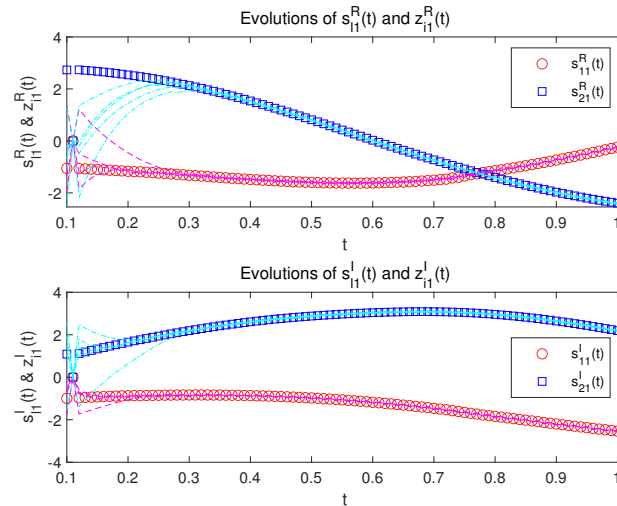
The states of  $s_l(t) (l = 1, 2)$  and  $z_i(t) (i = 1, 2, \dots, 9)$  are shown in Figures 4 and 5 under the controller (14). The red circle represents  $s_1(t)$ , the blue square is  $s_2(t)$ . The magenta double lines indicate neurons 1-4 belonging to the first cluster and all the dot dash lines in cyan indicate neurons 5-9 belonging to the second cluster. Obviously, they can achieve the FTCS, and the settling time is less than  $t^*$  in practice. The error systems are shown in Figure 6. In this figure, all magenta double lines represent the neurons belonging to the first cluster, and blue dotted lines represent the neurons belonging to the second cluster. As time goes by, the systematic errors tend to 0.



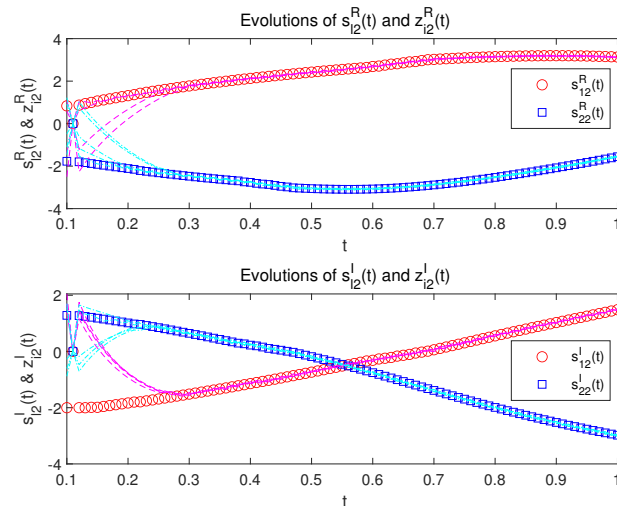
**Figure 3.** The topology of 9 neurons of the network (6) in Example 1.

The following is the optimization part of the control parameters. There are some parameters for the PSO algorithm. Let  $M = 50, N_m = 200, c_1 = c_2 = 2$ . Figure 7 describes the value of the fitness function (target function)  $J$  corresponding to the global optimal solution  $G_{best}$  during each iteration. Along with the increasing iteration times, the value of  $J$  becomes lower and lower. One can obtain the optimized control parameters, and the relevant  $J$  is 573.7228 in this simulation.



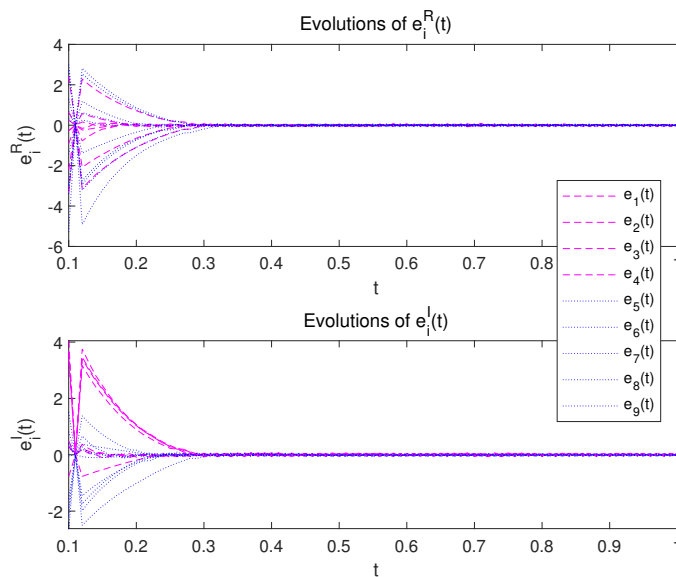


**Figure 4.** The stable evolution of the first dimensional state of the leaders  $s_{11}(t)$  and follower-neurons  $z_{i1}(t)$  in Example 1,  $l = 1, 2, i = 1, 2, \dots, 9$ .

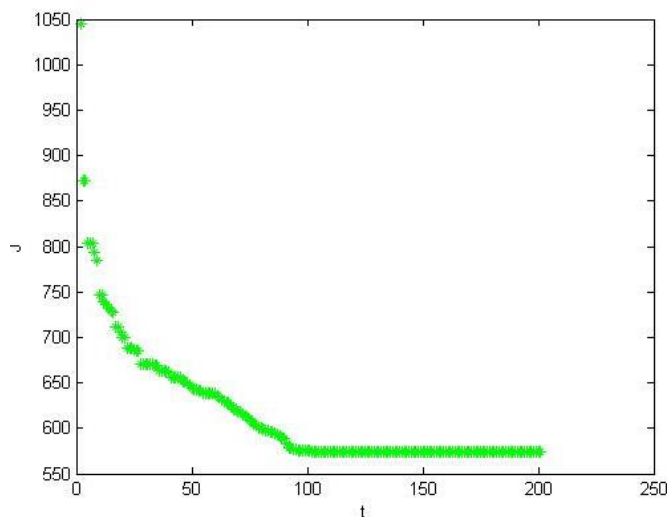


**Figure 5.** The stable evolution of the second dimensional state of the leaders  $s_{12}(t)$  and follower-neurons  $z_{i2}(t)$  in Example 1,  $l = 1, 2, i = 1, 2, \dots, 9$ .

There is a comparison experiment. The control parameters with and without the provided optimization algorithm are stated in Table 2. Most papers only give the range of control parameters, that is to say, the control parameters need to satisfy the conditions of Theorem 1. This method is expressed by the Normal method. The control gains is selected as  $k_1^1 = 9.87, k_2^1 = 10.17, k_3^1 = 10.47, k_4^1 = 9.87, k_5^1 = 9.87, k_6^1 = 9.87, k_7^1 = 10.47, k_8^1 = 9.87, k_9^1 = 10.17, k_1^2 = 4.84, k_2^2 = 4.6$ , in this method, which satisfy the Theorem 1, and the sum of ISE and control energy is 691.2507. This paper proposes an optimization method of control parameters, which is called the optimization method. The control parameters obtained according to the algorithm proposed in this paper is  $k_1^1 = 10.6886, k_2^1 = 1.3105 \times 10^{-4}, k_3^1 = 8.6932, k_4^1 = 8.2978, k_5^1 = 9.6064, k_6^1 = 8.4519, k_7^1 = 0.0408, k_8^1 = 10.643, k_9^1 = 10.355, k_1^2 = 5.6879, k_2^2 = 3.7145$  and the optimization objective is 573.7228. The control parameters and the provided optimization algorithm can obtain a lower sum of control energy and ISE index. Therefore, this optimization method can guide the selection of the control parameters.



**Figure 6.** The stable evolutions of 9 neurons’ errors  $e_i(t)$  in Example 1,  $i = 1, 2, \dots, 9$ .



**Figure 7.** The evolution of optimization target function  $J$  in Example 1.

**Remark 5.** The control matrix  $K^1$  can let the matrix  $\Xi_1 = I_N \otimes (\hat{C} - L\bar{A}) - \sigma\bar{G} \otimes \Gamma + K^1 \otimes I_n$  in Equation (16) is a diagonally-dominant matrix. At the same time,  $K^1$  has the minimum norm in the normal method of the comparison experiment. The selection of control matrices in [32] is similar to the normal method in this paper. Since the optimization of the controller is not considered in [32], the controller is not the minimum norm matrix that can make the matrix of synchronization condition is a diagonally dominant matrix in the simulation. However, the proposed optimization method can give better control parameters than the parameters via the normal method. That is to say, the proposed optimization method is better than an arbitrary selection within a certain range.

**Remark 6.** In this example, the “tanh” function is used as an activation function. In fact, all functions which satisfy Assumption 2 can be used as the activation function.

**Table 2.** Results with different control parameters in Example 1.

Method	Control	Parameters	$J$		
Normal	$k_i^1, i = 1 \rightarrow 3$	9.87	10.17	10.47	691.2507
	$k_i^1, i = 4 \rightarrow 6$	9.87	9.87	9.87	
	$k_i^1, i = 7 \rightarrow 9$	10.47	9.87	10.17	
	$k_i^2, i = 1, 2$	4.84	4.6		
Optimization	$k_i^1, i = 1 \rightarrow 3$	10.6886	$1.3105 \times 10^{-4}$	8.6932	573.7228
	$k_i^1, i = 4 \rightarrow 6$	8.2978	9.6064	8.4519	
	$k_i^1, i = 7 \rightarrow 9$	0.0408	10.643	10.355	
	$k_i^2, i = 1, 2$	5.6879	3.7145		

### 5.2. Example 2

There is a comparison experiment with [31] to further illustrate the advantage of the optimization method. The studied model is the same with Example 1 in [31]. The leader's state of each cluster is

$$D_{0,t}^\alpha s_l(t) = -Cs_l(t) + Bf(s_l(t)), \quad (49)$$

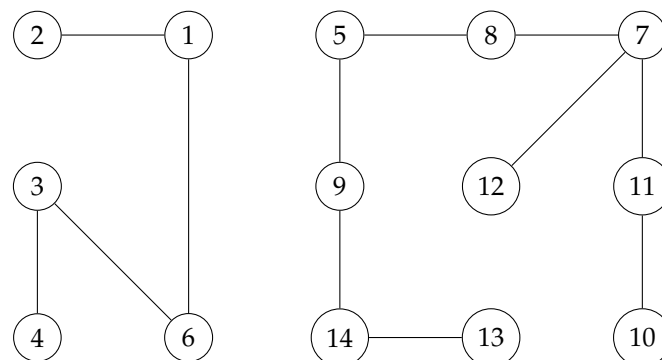
where  $l = 1, 2$ .

The CNN without memristor is described as follows:

$$D_{0,t}^\alpha z_i(t) = -Cz_i(t) + Bf(z_i(t)) + \sum_{j=1}^N g_{ij}h(z_j(t)) - d_l e_i(t) - \theta_l [e_i(t)], \quad (50)$$

where  $N = 14, \Lambda_1 = \{1, 2, \dots, 7\}, \Lambda_2 = \{8, 9, \dots, 14\}$ . When  $i \in \Lambda_1, l = 1$ , otherwise,  $l = 2$ .  $f(s(t)) = \tanh(s^R(t)) + \tanh(s^I(t))\mathbf{i}$ ,  $h(s(t)) = 0.1 \times \frac{1 - \exp(-s(t)^R)}{1 + \exp(-s(t)^R)} + \frac{0.1}{1 + \exp(-s(t)^I)}\mathbf{i}$ .

The topology of the network (50) is in Figure 8. If neurons  $i$  and  $j$  can connect with each other,  $g_{ij} = 1$ , otherwise,  $g_{ij} = 0$ , besides,  $g_{ii} = -\sum_{j=1}^n g_{ij}$ . The parameters are identical to Case 2 of Example 1 in [31], such as  $\alpha = 0.7, d_1 = 9, d_2 = 8, \theta_1 = \theta_2 = 5$ . The matrices  $C$  and  $B$ , and the initial value of  $s_l(t)$  are omitted here, which can be found in [31]. In [31], the settling time is 7.6398, and by calculation, the value of  $J$  in Equation (39) is  $3.7505 \times 10^3$ . We replaced the synchronization conditions of the optimization model in Section 4.1 with the synchronization conditions of Theorem 2 in [31]. The parameters of the optimization algorithm are  $M = 50, N_m = 100, c_1 = c_2 = 1$ , and we solved the optimization model with the step in Section 4.2. Then we found the optimal control parameters are  $d_1 = 13.8477, d_2 = 13.8515, \theta_1 = 2.5530, \theta_2 = 2.5525$ . Accordingly, the settling time is 7.5698, and the value of the fitness function  $J$  is  $2.8144 \times 10^3$ . The results of the comparison experiment are shown in Table 3 detailedly. Obviously, the settling times are close, but the values of  $J$  are  $2.8144 \times 10^3 < 3.7505 \times 10^3$ , which substantiate the proposed optimization method can help to select better control parameters, which can save control resources greatly.

**Figure 8.** The topology of 14 neurons of the network (50) in Example 2.

**Remark 7.** A lot of papers [13,14,18,20,21,23,31–33,37] only pay attention to the synchronization of the networks, they can give the range of control parameters, but our optimization method can select the optimal control parameters and obtain an energy-efficient controller.

**Table 3.** Results of comparison experiment in Example 2.

Method	$d_1$	$d_2$	$\theta_1$	$\theta_2$	$t^*$	$J$
[31]	9	8	5	5	7.6398	$3.7505 \times 10^3$
Optimization	13.8477	13.8515	2.5530	2.5525	7.5698	$2.8144 \times 10^3$

## 6. Conclusions

This paper investigates a coupled neural network based on a memristor, specifically a FOCV system with a time delay. The FTCS of these networks is analyzed, with neurons divided into different clusters, each following distinct leaders. A controller using the CVSF is designed to achieve FTCS without decomposition. Additionally, an optimization model solved by PSO is developed, guiding the selection of control parameters. The simulation results verify the proposed theorem and optimization method. This work successfully achieves FTCS for fractional-order complex systems. The advantage of the proposed theorem is that the FTCS of FOCV neural networks can be achieved without dividing the complex values into real and imaginary parts. Additionally, the model studied in this paper is more extensive, and the special case of the theorem is the conclusion in other literature, which is challenging. The advantage of the proposed algorithm is that the method of selecting control parameters can be given on the basis of the theorem, which makes up for the lack of research. However, the settling time of FTCS is influenced by the systems' initial values, which is difficult to obtain sometimes. Therefore, achieving fixed-time synchronization, where the settling time is independent of initial values, remains an area for future research, particularly for FOCV systems, which are not yet well-explored. That will be our future study.

**Author Contributions:** Writing—original draft preparation and funding, Q.C.; writing—review and editing, Q.C. and R.W.; supervision, Y.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (Grant No. 62303342).

**Data Availability Statement:** All data included in this study are available.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Ighamdi, F.A.; Almanaseer, H.; Jaradat, G.; Jaradat, A.; Alsmadi, M.K.; Jawarneh, S.; Almurayh, A.S.; Alqurni, J.; Alfagham, H. Multilayer perceptron neural network with arithmetic optimization algorithm-based feature selection for cardiovascular disease prediction. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 987–1008. [[CrossRef](#)]
2. Luo, N.; Xu, D.; Xing, B.; Yang, X.; Sun, C. Principles and applications of convolutional neural network for spectral analysis in food quality evaluation: A review. *J. Food Compos. Anal.* **2024**, *128*, 105996. [[CrossRef](#)]
3. Zhao, B.; Cao, X.; Zhang, W.; Liu, X.; Miao, Q.; Li, Y. CompNET: Boosting image recognition and writer identification via complementary neural network post-processing. *Pattern Recognit.* **2025**, *157*, 110880 [[CrossRef](#)]
4. Park, J.H. *Recent Advances in Control Problems of Dynamical Systems and Networks*; Springer-Nature: Cham, Switzerland, 2021.
5. Lai, Q.; Yang, L.; Hu, G.; Guan, Z.-H.; Iu, H.H.-C. Constructing multiscroll memristive neural network with local activity memristor and application in image encryption. *IEEE Trans. Cybern.* **2024**, *54*, 4039–4048. [[CrossRef](#)]
6. Liu, H.; Cheng, J.; Cao, J.; Katib, I. Preassigned-time synchronization for complex-valued memristive neural networks with reaction–diffusion terms and Markov parameters. *Neural Netw.* **2024**, *169*, 520–531 [[CrossRef](#)]

7. Zhang, Y.; Zhou, L. State estimation for proportional delayed complex-valued memristive neural networks. *Inf. Sci.* **2024**, *680*, 121150. [[CrossRef](#)]
8. Chang, Q.; Park, J.H.; Yang, Y. The optimization of control parameters: Finite-time bipartite synchronization of memristive neural networks with multiple time delays via saturation function. *IEEE Trans. Neural Networks Learn. Syst.* **2022**, *34*, 7861–7872. [[CrossRef](#)]
9. Aguirre, F.; Sebastian, A.; Le Gallo, M.; Song, W.; Wang, T.; Yang, J.J.; Lu, W.; Chang, M.-F.; Ielmini, D.; Yang, Y.; et al. Hardware implementation of memristor-based artificial neural networks. *Nat. Commun.* **2024**, *15*, 1974. [[CrossRef](#)]
10. Zhang, G.; Wen, S. New approximate results of fixed-time stabilization for delayed inertial memristive neural networks. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 3428–3432. [[CrossRef](#)]
11. Deng, Q.; Wang, C.; Lin, H. Memristive Hopfield neural network dynamics with heterogeneous activation functions and its application. *Chaos Solitons Fractals* **2024**, *178*, 114387. [[CrossRef](#)]
12. Li, Y.; Chen, Y.; Podlubny, I. Mittag–Leffler stability of fractional order nonlinear dynamic systems. *Automatica* **2009**, *45*, 1965–1969. [[CrossRef](#)]
13. Panda, S.K.; Nagy, A.; Vijayakumar, V.; Hazarika, B. Stability analysis for complex-valued neural networks with fractional order. *Chaos Solitons Fractals* **2023**, *175*, 114045. [[CrossRef](#)]
14. Feng, L.; Yu, J.; Hu, C.; Yang, C.; Jiang, H. Nonseparation method-based finite/fixed-time synchronization of fully complex-valued discontinuous neural networks. *IEEE Trans. Cybern.* **2020**, *51*, 212–3223. [[CrossRef](#)]
15. Chang, Q.; Park, J.H.; Yang, Y.; Wang, F. Finite-Time multiparty synchronization of T–S fuzzy coupled memristive neural networks with optimal event-triggered control. *IEEE Trans. Fuzzy Syst.* **2022**, *31*, 2545–2555. [[CrossRef](#)]
16. Sorrentino, F.; Pecora, L.M.; Hagerstrom, A.M.; Murphy, T.E.; Roy, R. Complete characterization of the stability of cluster synchronization in complex dynamical networks. *Sci. Adv.* **2016**, *2*, e1501737. [[CrossRef](#)] [[PubMed](#)]
17. Rossa, F.D.; Pecora, L.; Blaha, K.; Shirin, A.; Klickstein, I.; Sorrentino, F. Symmetries and cluster synchronization in multilayer networks. *Nat. Commun.* **2020**, *11*, 3179. [[CrossRef](#)]
18. Zhou, L.; Tan, F.; Yu, F.; Liu, W. Cluster synchronization of two-layer nonlinearly coupled multiplex networks with multi-links and time-delays. *Neurocomputing* **2019**, *359*, 264–275. [[CrossRef](#)]
19. Tang, Z.; Park, J.H.; Shen, H. Finite-time cluster synchronization of lur’e networks: A nonsmooth approach. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1213–1224. [[CrossRef](#)]
20. Zhang, Z.; Liu, X.; Zhou, D.; Lin, C.; Chen, J.; Wang, H. Finite-time stabilizability and instabilizability for complex-valued memristive neural networks with time delays. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 2371–2382. [[CrossRef](#)]
21. Chen, J.; Chen, B.; Zeng, Z. Global asymptotic stability and adaptive ultimate mittag-leffler synchronization for a fractional-order complex-valued memristive neural networks with delays. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2519–2535. [[CrossRef](#)]
22. Li, X.; Wu, H.; Cao, J. Synchronization in finite time for variable-order fractional complex dynamic networks with multi-weights and discontinuous nodes based on sliding mode control strategy. *Neural Netw.* **2021**, *139*, 335–347. [[CrossRef](#)] [[PubMed](#)]
23. Duan, L.; Shi, M.; Huang, C.; Fang, X. Synchronization in finite-/fixed-time of delayed diffusive complex-valued neural networks with discontinuous activations. *Chaos Solitons Fractals* **2021**, *142*, 110386. [[CrossRef](#)]
24. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN’95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
25. Tan, Y.; Yuan, Y.; Xie, X.; Tian, E.; Liu, J. Observer-based event-triggered control for interval type-2 fuzzy networked system with network attacks. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 2788–2798. [[CrossRef](#)]
26. Liu, B.; Liu, T.; Xiao, P. Dynamic event-triggered intermittent control for stabilization of delayed dynamical systems. *Automatica* **2023**, *149*, 110847. [[CrossRef](#)]
27. Li, X.; Liu, W.; Gorbachev, S.; Cao, J. Event-triggered impulsive control for input-to-state stabilization of nonlinear time-delay systems. *IEEE Trans. Cybern.* **2023**, *54*, 2536–2544. [[CrossRef](#)]
28. Liu, Z.; Gao, H.; Yu, X.; Lin, W.; Qiu, J.; Rodríguez-Andina, J.J.; Qu, D. B-spline wavelet neural-network-based adaptive control for linear-motor-driven systems via a novel gradient descent algorithm. *IEEE Trans. Ind. Electron.* **2023**, *71*, 1896–1905. [[CrossRef](#)]
29. Wang, J.; Ru, T.; Shen, H.; Cao, J.; Park, J.H. Finite-time  $L_2 - L_\infty$  synchronization for semi-markov jump inertial neural networks using sampled data. *IEEE Trans. Network Sci. Eng.* **2021**, *8*, 163–173. [[CrossRef](#)]
30. Park, J.H.; Lee, T.H.; Liu, Y.; Chen, J. *Dynamic Systems with Time Delays: Stability and Control*; Springer-Nature: Singapore, 2019.
31. Yang, S.; Hu, C.; Yu, J.; Jiang, H. Finite-time cluster synchronization in complex-variable networks with fractional-order and nonlinear coupling. *Neural Netw.* **2021**, *135*, 212–224. [[CrossRef](#)] [[PubMed](#)]
32. Liu, P.; Zeng, Z.; Wang, J. Asymptotic and finite-time cluster synchronization of coupled fractional-order neural networks with time delay. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 4956–4967. [[CrossRef](#)]
33. Wang, L.; Song, Q.; Liu, Y.; Zhao, Z.; Alsaadi, F.E. Finite-time stability analysis of fractional-order complex-valued memristor-based neural networks with both leakage and time-varying delays. *Neurocomputing* **2017**, *245*, 86–101. [[CrossRef](#)]

34. Yu, T.; Cao, J.; Rutkowski, L.; Luo, Y.-P. Finite-time synchronization of complex-valued memristive-based neural networks via hybrid control. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3938–3947. [[CrossRef](#)] [[PubMed](#)]
35. Zhu, X.; Tang, Z.; Feng, J.; Park, J.H. Aperiodically intermittent event-triggered pinning control on cluster synchronization of directed complex networks. *ISA Trans.* **2023**, *138*, 281–290. [[CrossRef](#)]
36. Zhang, J.; Ma, Z.; Li, X.; Qiu, J. Cluster Synchronization in Delayed Networks With Adaptive Coupling Strength via Pinning Control. *Front. Phys.* **2020**, *8*, 235. [[CrossRef](#)]
37. Hou, T.; Yu, J.; Hu, C.; Jiang, H. Finite-time synchronization of fractional-order complex-variable dynamic networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 4297–4307. [[CrossRef](#)]
38. Podlubny, I. *Fractional Differential Equations*; Academic Press: London, UK, 1999.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.