



The Optimization of Synchronization Control Parameters for Fractional-Order Delayed Memristive Neural Networks Using SIWPSO

Qi Chang² · Aihua Hu¹ · Yongqing Yang¹ · Li Li¹

Published online: 3 December 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

The paper mainly deals with the optimization of synchronization for fractional-order memristive neural networks (FOMNNs) with a time delay. Based on synchronization conditions, an optimization model for control parameters is designed and computed. It's significative to design an appropriate controller which can synchronize the drive FOMNNs and response FOMNNs. Based on the proposed controller, some synchronization conditions of FOMNNs can be obtained with the help of the linear matrix inequality, along with fractional-order Lyapunov methods and matrix analysis. The optimal model of control parameters includes a target function and some constraints. The target function is the minimal sum of control energy and integral square error index. The constraint conditions choose the sufficient conditions for synchronization of FOMNNs. The optimization model is difficult to compute but can be solved by means of the stochastic inertia weight particle swarm optimization algorithm. A simulation is provided to verify the validity of the proposed theoretical results.

Keywords Fractional-order · Memristive neural networks · Optimal control · SIWPSO algorithm

1 Introduction

As the extension of the integer-order calculus, the basic theory of fractional differential and integral calculus were studied in early period. With the improvement of computing technology and the realization of the analog circuit, the fractional calculus has been widely applied in physics and engineering in recent years. One of the advantages of fractional calculus is that it's

This work was jointly supported by the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK20161126, BK20170171, BK20181342, and the Postgraduate Research and Practice Innovation Program of Jiangnan University under Grant No. JNKY19_042.

✉ Yongqing Yang
yongqingyang@jiangnan.edu.cn

¹ School of Science, Wuxi Engineering Research Center for Biocomputing, Jiangnan University, Wuxi 214122, People's Republic of China

² School of IoT Engineering, Jiangnan University, Wuxi 214122, People's Republic of China

an excellent mathematical tool for describing the memory and genetic characteristic of many novel materials and special processes. Furthermore, compared to the integer-order model, the fractional-order model has more degrees of freedom and infinite memories. Based on the salient properties of fractional calculus, many practical systems described by fractional order models could reflect the characteristics of the system more accurately than integer order models. As a result, fractional calculus has been applied in many territories, such as quantum mechanics [1], viscoelasticity [2], control engineering, robotics and chaotic synchronization [3].

Neural network is an intricate system formed by high interconnection of a large number of neurons originally. Scientists have constructed artificial neural networks (neural networks for short) after understanding the biological neural networks. Neural networks [4–7] have attracted wide interest in many domains, such as pattern recognition, artificial intelligence and signal processing. Since the discovery of physical devices with memristive characteristics in HP LABS in 2008, the memristive neural networks (MNNs) model has become a new hot topic [8,9]. A neural network consists of neurons (nodes) and synapses (connections between nodes). In order to train the neural network to complete a task, it needs to be “fed” a large number of questions and corresponding answers. Once trained, neural networks can be tested without knowing the answer. The training process requires a lot of manpower, resources and time. It is also very expensive. However, by using memristors, most of the expensive training processes can be avoided. The memristors can provide memory ability to the networks as well. The memory storage capacity will be significantly improved due to the application of memristive neural networks in associative memories. The memristors which have different history-dependent resistors, make the memristive neural networks have more complicated and fruitful dynamics than traditional neural networks.

Synchronization is a significant research subject of chaotic systems. MNN is one kind of chaotic systems. Hence, many scholars from in and abroad pay attentions to the synchronization of MNNs. Global anti-synchronization of MNNs was analyzed by Zhang et al. [10]. Exponential synchronization of Cohen–Grossberg neural networks with memristors was analyzed by Yang et al. [11]. Robust synchronization of multiple MNNs was analyzed by Yang et al. [12]. The MNNs cannot be spontaneously synchronized upon most occasions. Thus, control schemes play an important role. There are many control strategies, such as adaptive control [13], intermittent control [14], sampled-data control [15], event-trigger [16] control and so on.

More and more attentions have been paid to the parameter optimization of synchronization controller [17,18]. A particle swarm optimization (PSO) algorithm was exploited by Kennedy and Eberhart. It is an evolutionary algorithm. The algorithm begins with some random solutions, then searches the optimum solution through iterations, and estimates the solution’s quality according to the fitness function in the PSO algorithm. The PSO algorithm has more simply rule compared with the rule of a genetic algorithm (GA). The crossover and mutation operation are needed in the GA, nevertheless, the PSO algorithm can do without them. The PSO algorithm is based on the current optimal solution to find the location of the global optimal solution. The algorithm has gained the attention of academia due to its advantages such as easy implementation, high precision computation and rapid convergence, which has already confirmed its advantages in dealing with real issues. The improved particle swarm optimization algorithms [19,20] have attracted much interest in the area of artificial intelligence. Stochastic inertia weight particle swarm optimization (SIWPSO) algorithm [21] is one of them which can find optimal solutions effectively.

A time delay [22] is inevitable in practical application. However, its existence will affect the networks to become unstable. Therefore, the discussion on the time delay is necessary when

the synchronization of FOMNNs is studied. Many articles have analyzed the synchronization of FOMNNs [23,24]. However, the optimization of controller parameters is not considered in these articles. Many articles about PID controller have applied the PSO algorithm to deal with the optimization problem [25,26]. The authors in [27] have analyzed the optimal synchronization of complex networks via PSO algorithm. Try the best of the authors' ability, there is no literature investigating the optimization of the FOMNNs by the SIWPSO algorithm before. In summary, the paper investigates the optimal synchronization of fractional-order delayed memristive neural networks along with SIWPSO.

The main contributions of this paper include three aspects. (1) The drive-response synchronization for fractional-order memristive neural networks with time delay is studied in this paper. For convenience, the model is described in compact forms. (2) Based on fractional-order Lyapunov methods and some inequations, synchronization conditions are shown in linear matrix inequality's (LMIs) form. The proposed methods apply equally to the synchronization problem for fractional-order delayed neural networks and fractional-order memristive neural networks. (3) To get a better controller with low control energy and integral square error (ISE) index, an optimization model is established to solve the optimal control parameters. The complicated model is solved by SIWPSO algorithm, which is an improved intelligent algorithm.

The rest of this paper is arranged as follows: some definitions, lemmas and assumptions are introduced in Sect. 2. The models of the FOMNNs are described comprehensively and distinctly in this section as well. Section 3 presents some achievements about the synchronization of the FOMNNs with a constant time delay. Simultaneously, two corollaries are also given in this section. One is the synchronization without memristor, and the other is without the time delay. Then, a parameter optimization of the controller is specifically designed in Sect. 4, which includes the optimization model and the SIWPSO algorithm. Finally, numerical examples and the conclusions are provided in Sects. 5 and 6 respectively.

Notations Let I_n represent the $n \times n$ identity matrix. Let T stand for matrix transposition. Let R, \mathbb{R}^n and $\mathbb{R}^{m \times n}$ respectively denote the set of real numbers, n-dimensional vectors and $m \times n$ real matrices. $\|x\|$ is the Euclidean norm for a vector $x \in \mathbb{R}^n$. $\|\Gamma\|$ represents the 2-norm of Γ for a matrix $\Gamma \in \mathbb{R}^{n \times n}$. A diagonal matrix can be expressed by $diag\{\dots\}$. If a matrix P is a symmetric positive (semi-positive) definite matrix, it will be expressed in terms of $P > 0(\geq 0)$. Finally, “ \star ” are used to represent the omitted symmetry parts of the symmetric block matrix.

2 Preliminary Knowledge and Model Description

This section gives some lemmas and definitions of fractional calculus which play important roles in next parts. Besides, the mathematical models of FOMNNs with a time delay are introduced.

2.1 Fractional Calculus

Definition 1 [28] When fractional order $\alpha \in \mathbb{R}^+$, the Caputo derivative with function $\omega(s)$ is defined as:

$$D_{t_0,t}^\alpha \omega(s) = D_{t_0,t}^{-(n-\alpha)} \frac{d^n}{dt^n} \omega(s) = \frac{1}{\Gamma(n-\alpha)} \int_{t_0}^t (s-\tau)^{n-\alpha-1} \omega^{(n)}(\tau) d\tau,$$

where $t \geq t_0$ and $n - 1 < \alpha < n \in \mathbb{Z}^+$, and $\Gamma(\cdot)$ is the Gamma function and $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt, t \geq t_0$. Specifically, $D_{t_0,t}^\alpha \omega(s) = \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^t (s - \tau)^{-\alpha} \omega'(\tau) d\tau$, when $0 < \alpha < 1$.

Remark 1 The Caputo derivative of constant functions is zero. Besides, the initial condition of Caputo differential operator is in the same form as that of integer order differential equations [29]. The Caputo derivative is more appropriate for a realistic model and is chosen in this paper.

Lemma 1 [30] *Let $f(t) \in \mathbb{R}^n$ is a vector of differentiable function, for any time instant $t \geq t_0$, the inequality holds*

$$D^\alpha [f^T(t) P f(t)] \leq 2 f^T(t) P D^\alpha f(t), \forall \alpha \in (0, 1],$$

where P is a symmetric positive definite constant matrix.

Lemma 2 [31] *Let $X, Z \in \mathbb{R}^n, \epsilon > 0$ is a constant, then*

$$X^T Z + Z^T X \leq \epsilon^{-1} X^T X + \epsilon Z^T Z.$$

Lemma 3 [32] *The linear matrix inequality*

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{12}^T & S_{22} \end{pmatrix} < 0,$$

if and only if conditions (i) or (ii) holds:

- (i) $S_{11} < 0, S_{22} - S_{12}^T S_{11}^{-1} S_{12} < 0,$
- (ii) $S_{22} < 0, S_{11} - S_{12}^T S_{22}^{-1} S_{12} < 0,$

where $S_{11} = S_{11}^T$ and $S_{22} = S_{22}^T$.

Lemma 4 [33] *Suppose that $\omega_1, \omega_2 : R \rightarrow R$ are continuous and strictly increasing. $\omega_1(s)$ and $\omega_2(s)$ are positive for $s > 0$, and $\omega_1(0) = \omega_2(0) = 0, \omega_1, \omega_2$ strictly increasing. If there exists a continuous and differentiable function $V : R \times \mathbb{R}^n \rightarrow R$ so that $\omega_1(\|x(t)\|) \leq V(t, x(t)) \leq \omega_2(\|x(t)\|)$ holds, and there exist two constants $\kappa > \beta > 0$ so that for any given $t_0 \in R$ the Caputo system $D^\alpha x(t) = f(t, x(t), x(t - \tau))$ satisfies*

$$D^\alpha V(t, x(t)) \leq -\kappa V(t, x(t)) + \beta \sup_{-\tau \leq \theta \leq 0} V(t + \theta, x(t + \theta))$$

for $t \geq t_0$, then the Caputo system $D^\alpha x(t) = f(t, x(t), x(t - \tau))$ will achieve global asymptotical stability.

Remark 2 It's difficult to directly synchronize fractional-order neural networks(FONNs) through common integer-order Lyapunov methods. However, Lemmas 1 and 4 are adopted to synchronize the fractional-order ones.

2.2 Model Description

The FOMNNs with a constant time delay is considered as follows:

$$D^\alpha x(t) = -Dx(t) + A(x(t))f(x(t)) + B^T(x(t))f(x(t - \tau)) + I \tag{1}$$

where $0 < \alpha < 1$, $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T \in \mathbb{R}^n$ is the neuron's state variable, $D = \text{diag}\{d_1, d_2, \dots, d_n\}$, $d_i > 0 (i = 1, 2, \dots, n)$ stands for the rate at which the i th neuron returns to the resting state, $f(x(t)) = (f_1(x_1(t)), f_2(x_2(t)), \dots, f_n(x_n(t)))^T$ is the activation function of neurons without the time delay. $f(x(t - \tau))$ is the activation function with the time delay, τ represents the constant time delay. $I = (I_1(t), I_2(t), \dots, I_n(t))^T \in \mathbb{R}^n$ denotes an external input vector. $A(x(t)) = [a_{ij}(x_i(t))]_{n \times n}$, $B^\tau(x(t)) = [b_{ij}^\tau(x_i(t))]_{n \times n}$ are the connection memristive weight matrices at time t and $t - \tau$:

$$a_{ij}(x_i(t)) = \frac{\tilde{R}_{ij}^1}{C_i} \text{sgn}_{ij}, \quad b_{ij}^\tau(x_i(t)) = \frac{\tilde{R}_{ij}^2}{C_i} \text{sgn}_{ij},$$

$$\text{sgn}_{ij} = \begin{cases} 1, & i \neq j, \\ -1, & i = j. \end{cases}$$

where \tilde{R}_{ij}^1 and \tilde{R}_{ij}^2 denote the memductance of memristors R_{ij}^1 and R_{ij}^2 , respectively. R_{ij}^1 donates the memristor between $f_j(x_j(t))$ and $x_i(t)$, and R_{ij}^2 donates the memristor between $f_j(x_j(t - \tau))$ and $x_i(t - \tau)$. On account of the current and voltage feature and the characteristics of memristor, a general mathematical model with the memristance is established as follows [34]:

$$a_{ij}(x_i(t)) = \begin{cases} \acute{a}_{ij}, & |x_i(t)| \leq T_i, \\ \grave{a}_{ij}, & |x_i(t)| > T_i, \end{cases} \quad b_{ij}^\tau(x_i(t)) = \begin{cases} \acute{b}_{ij}^\tau, & |x_i(t)| \leq T_i, \\ \grave{b}_{ij}^\tau, & |x_i(t)| > T_i, \end{cases}$$

where $T_i > 0$ are switching jumps, weights \acute{a}_{ij} , \grave{a}_{ij} , \acute{b}_{ij}^τ and \grave{b}_{ij}^τ are all constants for $1 \leq i, j \leq n$.

The Filippov solution [35] for all systems is considered on account of the discontinuity of $a_{ij}(x_i(t))$ and $b_{ij}^\tau(x_i(t))$ in this paper. Denote $\bar{a}_{ij} = \max\{\acute{a}_{ij}, \grave{a}_{ij}\}$, $\underline{a}_{ij} = \min\{\acute{a}_{ij}, \grave{a}_{ij}\}$, $\bar{b}_{ij}^\tau = \max\{\acute{b}_{ij}^\tau, \grave{b}_{ij}^\tau\}$, $\underline{b}_{ij}^\tau = \min\{\acute{b}_{ij}^\tau, \grave{b}_{ij}^\tau\}$, $a_{ij} = \frac{1}{2}(\bar{a}_{ij} + \underline{a}_{ij})$, $b_{ij}^\tau = \frac{1}{2}(\bar{b}_{ij}^\tau + \underline{b}_{ij}^\tau)$, $\tilde{a}_{ij} = \frac{1}{2}(\bar{a}_{ij} - \underline{a}_{ij})$ and $\tilde{b}_{ij}^\tau = \frac{1}{2}(\bar{b}_{ij}^\tau - \underline{b}_{ij}^\tau)$. According to some equation transformations [36,37] and the fractional differential methods [38], the FOMNNs (1) can be rewritten to the following system:

$$D^\alpha x(t) = -Dx(t) + (A + M\Lambda_1(t)E)f(x(t)) + (B^\tau + \hat{M}\Lambda_2(t)\hat{E})f(x(t - \tau)) + I. \tag{2}$$

Throughout this paper, the FOMNNs (2) is considered as the drive FOMNNs, the response FOMNNs with control is as described by:

$$D^\alpha y(t) = -Dy(t) + (A + M\Lambda_3(t)E)f(y(t)) + (B^\tau + \hat{M}\Lambda_4(t)\hat{E})f(y(t - \tau)) + u(t) + I \tag{3}$$

where $u(t) = (u_1(t), u_2(t), \dots, u_n(t))^T$ is the controller to be designed. $A = (a_{ij})_{n \times n}$, $B^\tau = (b_{ij}^\tau)_{n \times n}$, $M = (\sqrt{\tilde{a}_{11}}\xi_1, \dots, \sqrt{\tilde{a}_{1n}}\xi_1, \dots, \sqrt{\tilde{a}_{n1}}\xi_n, \dots, \sqrt{\tilde{a}_{nn}}\xi_n)_{n \times n^2}$, $E = (\sqrt{\tilde{a}_{11}}\xi_1, \dots, \sqrt{\tilde{a}_{1n}}\xi_n, \dots, \sqrt{\tilde{a}_{n1}}\xi_1, \dots, \sqrt{\tilde{a}_{nn}}\xi_n)_{n^2 \times n}$, $\hat{M} = (\sqrt{\tilde{b}_{11}^\tau}\xi_1, \dots, \sqrt{\tilde{b}_{1n}^\tau}\xi_1, \dots, \sqrt{\tilde{b}_{n1}^\tau}\xi_n, \dots, \sqrt{\tilde{b}_{nn}^\tau}\xi_n)_{n \times n^2}$, $\hat{E} = (\sqrt{\tilde{b}_{11}^\tau}\xi_1, \dots, \sqrt{\tilde{b}_{1n}^\tau}\xi_n, \dots, \sqrt{\tilde{b}_{n1}^\tau}\xi_1, \dots, \sqrt{\tilde{b}_{nn}^\tau}\xi_n)_{n^2 \times n}$, $\xi_i \in \mathbb{R}^n$ is the column vector which the i th element is 1 and others are all 0, $\{[\Lambda_k(t)]_{n^2 \times n^2} = \text{diag}\{\theta_{11}^k(t), \dots, \theta_{1n}^k(t), \dots, \theta_{n1}^k(t), \dots, \theta_{nn}^k(t)\} : |\theta_{ij}^k| \leq 1, 1 \leq i, j \leq n, k = 1, 2, 3, 4\}$. It's easy to obtain that $\Lambda_k^T(t)\Lambda_k(t) \leq I (k = 1, 2, 3, 4)$.

Let $e(t) = y(t) - x(t)$, the error dynamical system is derived from (2) and (3):

$$D^\alpha e(t) = -De(t) + (A + \Delta A)g(e(t)) + (B^\tau + \Delta B^\tau)g(e(t - \tau)) + \Sigma(t) + u(t) \tag{4}$$

where $\Delta A = M\Lambda_3(t)E$, $\Delta B = \widehat{M}\Lambda_4(t)\widehat{E}$, $g(e(t)) = f(y(t)) - f(x(t))$, $g(e(t - \tau)) = f(y(t - \tau)) - f(x(t - \tau))$, $\Sigma(t) = [M(\Lambda_3(t) - \Lambda_1(t))E]f(x(t)) + [\widehat{M}(\Lambda_4(t) - \Lambda_2(t))\widehat{E}]f(x(t - \tau))$. The initialization of (4) is $e(s) = \Psi(s) \in \mathcal{C}([-\tau, 0], \mathbb{R}^n)$.

There are some assumptions required for the following research:

Assumption 1 The activation functions f_j are Lipschitz continuous on R , i.e.,

$$|f_j(u) - f_j(v)| \leq L_j|u - v|$$

where $u, v \in R$, $j = 1, 2, \dots, n$ and constant $L_j > 0$.

Assumption 2 Constants M_j exist so that $|f_j(z)| \leq M_j$ for $\forall z \in \mathbb{R}$ and $j = 1, 2, \dots, n$.

When $|x_i(t)| \leq T_i$ at time t , the state $y_i(t)$ will not be confirmed whether $|y_i(t)| \leq T_i$ or $|y_i(t)| \geq T_i$ due to their different initial value. As a result, it's easy to realize that $\Lambda_3(t) \neq \Lambda_1(t)$, $\Lambda_4(t) \neq \Lambda_2(t)$. Hence, the linear feedback controller such as those in [39,40] can not be used to synchronize the FOMNNs firsthand. In addition, the term $\Sigma(t)$ of error system (4) can be treated as an external perturbation under some assumption conditions. Based on these, the following controller is designed:

$$u(t) = -Ke(t) - \Gamma \text{sgn}(e(t)), \tag{5}$$

where K is a real matrix, $\Gamma = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_n\}$, K and Γ are gain matrices to be confirmed. $e(t) = (e_1(t), e_2(t), \dots, e_n(t))^T$, $\text{sgn}(\cdot)$ is the standard sign function.

3 Main Result: Synchronization Criteria

Based on the proposed controller (5), some novel sufficient criteria are obtained to synchronize the FOMNNs (2) and (3). The definition and theorems of synchronization are present for FOMNNs in this section. Some corollaries are given as well.

Definition 2 The drive-response synchronization of the two FOMNNs (2) and (3) is said to be achieved, if a suitable controller $u(t)$ is designed,

$$\lim_{t \rightarrow \infty} \|y(t) - x(t)\| = 0$$

holds for any initial values.

Theorem 1 Suppose that Assumptions 1 and 2 hold, for given constants $0 < \beta < \kappa$, under the proposed controller (5), with $\gamma_i = \sum_{j=1}^n (|\dot{a}_{ij} - \dot{a}_{ij}| + |\dot{b}_{ij}^\tau - \dot{b}_{ij}^\tau|)M_j$, the FOMNNs (2) and (3) can reach global asymptotical synchronization, if there exists a matrix Y , some positive constants ϵ, ρ_1, ρ_2 and a positive definite matrix Q so that the following LMIs (6) and (7) are satisfied:

$$\Pi = \begin{pmatrix} \Pi_{11} & QB & QM & 0 \\ * & -\epsilon I_n & 0 & QE\widehat{M} \\ * & * & -\rho_1 I_{n^2 \times n^2} & 0 \\ * & * & * & -\rho_2 I_{n^2 \times n^2} \end{pmatrix} < 0, \tag{6}$$

$$\epsilon L^T L - \beta Q < 0 \tag{7}$$

where $\Pi_{11} = -QD - D^T Q - Y - Y^T + QAL + L^T A^T Q + \kappa Q + \rho_1 L^T E^T EL + \rho_2 \widehat{E}^T \widehat{E}$, $K = Q^{-1}Y$.

Proof Construct the Lyapunov function as follows:

$$V(t) = e^T(t)Qe(t)$$

where $Q = \text{diag}\{q_1, q_2, \dots, q_n\}$.

Taking the time derivative of $V(t)$ along the error system and applying Lemma 1 and Assumption 1, one can get

$$\begin{aligned} D^\alpha V(t) &\leq 2e^T(t)QD^\alpha e(t) \\ &= 2e^T(t)Q[-De(t) + (A + \Delta A)g(e(t)) + (B^\tau + \Delta B^\tau)g(e(t - \tau)) \\ &\quad + \Sigma(t) - Ke(t) - \Gamma\omega(t)] \\ &\leq e^T(t)[-QD - D^T Q - QK - K^T Q]e(t) + 2e^T(t)Q(A + \Delta A)g(e(t)) \\ &\quad + \|2e^T(t)Q(B^\tau + \Delta B^\tau)g(e(t - \tau))\| + 2e^T(t)Q[\Sigma(t) - \Gamma\omega(t)] \\ &\leq e^T(t)[-QD - D^T Q - QK - K^T Q]e(t) \\ &\quad + e^T(t)[Q(A + \Delta A)L + L^T(A + \Delta A)^T Q]e(t) \\ &\quad + \|e^T(t)Q(B^\tau + \Delta B^\tau)Le(t - \tau) + e^T(t - \tau)L^T(B^\tau + \Delta B^\tau)^T Qe(t)\| \\ &\quad + 2e^T(t)Q[\Sigma(t) - \Gamma\omega(t)] \end{aligned}$$

where $A + \Delta A > 0$, $L = \text{diag}\{L_1, L_2, \dots, L_n\}$, $\omega(t) = (\omega_1(t), \omega_2(t), \dots, \omega_n(t))^T, i = 1, 2, \dots, n$,

$$\omega_i(t) = \begin{cases} \text{sgn}(e_i(t)), & \text{if } e_i(t) \neq 0, \\ 0, & \text{if } e_i(t) = 0. \end{cases}$$

According to Lemma 2, it holds for any $t \in [0, \infty)$ that

$$\begin{aligned} &\|e^T(t)Q(B^\tau + \Delta B^\tau)Le(t - \tau) + e^T(t - \tau)L^T(B^\tau + \Delta B^\tau)^T Qe(t)\| \\ &\leq \frac{1}{\epsilon}e^T(t)Q(B^\tau + \Delta B^\tau)(B^\tau + \Delta B^\tau)^T Qe(t) + \epsilon e^T(t - \tau)L^T Le(t - \tau). \end{aligned}$$

Noticing that $2\tilde{a}_{ij} = |\dot{a}_{ij} - \ddot{a}_{ij}|$, $2\tilde{b}_{ij}^\tau = |\dot{b}_{ij}^\tau - \ddot{b}_{ij}^\tau|$, $\gamma_i = \sum_{j=1}^n (|\dot{a}_{ij} - \ddot{a}_{ij}| + |\dot{b}_{ij}^\tau - \ddot{b}_{ij}^\tau|)M_j = \varphi_i$, along with the Assumption 2, one can get that

$$2e^T(t)Q[\Sigma(t) - \Gamma\omega(t)] \leq 2 \sum_{i=1}^n |e_i(t)|q_i(\varphi_i - \gamma_i) = 0.$$

From the above, differentiating $V(t)$ along the solutions of (4) becomes

$$\begin{aligned} D^\alpha V(t) &\leq e^T(t)[-QD - D^T Q - QK - K^T Q + Q(A + \Delta A)L + L^T(A + \Delta A)^T Q \\ &\quad + \frac{1}{\epsilon}Q(B^\tau + \Delta B^\tau)(B^\tau + \Delta B^\tau)^T Q]e(t) + \epsilon e^T(t - \tau)L^T Le(t - \tau) \quad (8) \\ &= e^T(t)\Xi e(t) + \epsilon e^T(t - \tau)L^T Le(t - \tau), \end{aligned}$$

where $\Xi = -QD - D^T Q - QK - K^T Q + Q(A + \Delta A)L + L^T(A + \Delta A)^T Q + \frac{1}{\epsilon}Q(B^\tau + \Delta B^\tau)(B^\tau + \Delta B^\tau)^T Q$.

Let

$$\Omega = \begin{pmatrix} \Omega_{11} & Q(B^\tau + \Delta B^\tau) \\ * & -\epsilon I_n \end{pmatrix}, \quad \Phi_1 = \begin{pmatrix} \Phi_{11} & Q B^\tau \\ * & -\epsilon I_n \end{pmatrix}, \quad \Phi_2 = \begin{pmatrix} \Phi'_{11} & Q \Delta B^\tau \\ * & 0_{n \times n} \end{pmatrix},$$

where $\Omega_{11} = -QD - D^T Q - QK - K^T Q + Q(A + \Delta A)L + L^T(A + \Delta A)^T Q + \kappa Q$, $\Phi_{11} = -QD - D^T Q - QK - K^T Q + QAL + L^T A^T Q + \kappa Q$, $\Phi'_{11} = Q \|\Delta A\|L + L^T \|\Delta A^T\|Q$. It's easy to see that $\Omega \leq \Phi_1 + \Phi_2$.

$$\begin{aligned} \Phi_2 &= \begin{pmatrix} Q \|\Delta A\|L + L^T \|\Delta A^T\|Q & Q \Delta B^\tau \\ * & 0_{n \times n} \end{pmatrix} \\ &= \begin{pmatrix} QM \|\Lambda_3(t)\|EL + L^T E^T \|\Lambda_3^T(t)\|M^T Q & Q \widehat{M} \|\Lambda_4(t)\| \widehat{E} \\ * & 0_{n \times n} \end{pmatrix}, \end{aligned}$$

Let

$$N_1 = (QM, 0_{n \times n^2})^T, N_2 = (0_{n \times n^2}, Q \widehat{M})^T, F_1 = (EL, 0_{n^2 \times n}), F_2 = (\widehat{E}, 0_{n^2 \times n}),$$

and according to Lemma 2, Φ_2 can be rewritten as follows:

$$\begin{aligned} \Phi_2 &= N_1 \|\Lambda_3(t)\| F_1 + F_1^T \|\Lambda_3^T(t)\| N_1^T + N_2 \|\Lambda_4(t)\| F_2 + F_2^T \|\Lambda_4^T(t)\| N_2^T \\ &\leq \rho_1^{-1} N_1 N_1^T + \rho_1 F_1^T F_1 + \rho_2^{-1} N_2 N_2^T + \rho_2 F_2^T F_2. \end{aligned}$$

By Lemma 3, the Theorem's condition (6) $\Pi < 0$ is equivalent to

$$\Phi_1 + \rho_1 F_1^T F_1 + \rho_2 F_2^T F_2 + \rho_1^{-1} N_1 N_1^T + \rho_2^{-1} N_2 N_2^T < 0,$$

which means $\Omega < 0$. Based on Lemma 3 again, $\Omega < 0$ means that

$$\begin{aligned} &-QD - D^T Q - QK - K^T Q + Q(A + \Delta A)L + L^T(A + \Delta A)^T Q \\ &+ \frac{1}{\epsilon} Q(B^\tau + \Delta B^\tau)(B^\tau + \Delta B^\tau)^T Q + \kappa Q < 0, \end{aligned}$$

that is

$$\Xi < -\kappa Q. \tag{9}$$

Using (7) and (9), the derivative of $V(t)$ (8) can be rewritten as:

$$D^\alpha \leq -\kappa V(t) + \beta V(t - \tau).$$

If the condition $\kappa > \beta > 0$ is combined, then the conditions of the Lemma 4 are all satisfied. As a consequence, error system (4) is asymptotically stable. It's not complicated to get that $e(t) \rightarrow 0$, when $t \rightarrow \infty$, that is to say, $\lim_{t \rightarrow \infty} \|y(t) - x(t)\| = 0$. On the basis of Definition 3, the drive FOMNNs (2) and response FOMNNs (3) can synchronize via the designed controller (5). This completes the proof. \square

Remark 3 The drive-response FOMNNs (2) and (3) can achieve synchronization by ingeniously designed controller (5) with $\gamma_i = \sum_{j=1}^n (|\hat{a}_{ij} - \hat{a}_{ij}| + |\hat{b}_{ij}^\tau - \hat{b}_{ij}^\tau|) M_j$. Underlying given constants κ, β and defined $Y = QK$, the synchronization criteria LMIs (6) and (7) are easy to be solved by the matlab toolbox.

A fractional-order drive-response neural networks with a constant time delay is established as follows:

$$D^\alpha x(t) = -Dx(t) + Af(x(t)) + B^\tau f(x(t - \tau)) + I \tag{10}$$

$$D^\alpha y(t) = -Dy(t) + Af(y(t)) + B^T f(y(t - \tau)) + u'(t) + I \tag{11}$$

The controller is reduced to

$$u'(t) = -Ke(t), \tag{12}$$

then the error system is taken as follows:

$$D^\alpha e(t) = -(D + K)e(t) + Ag(e(t)) + B^T g(e(t - \tau)) \tag{13}$$

Based on controller (13), the synchronization criteria for fractional neural networks(FNNs) is obtained in Corollary 1.

Corollary 1 *Suppose that Assumption 1 holds, for given constants $0 < \beta < \kappa$, under the proposed controller (12), the FNNs (10) and (11) can globally and asymptotically synchronize, if there exists a matrix Y , a positive constant ϵ and a positive definite matrix $Q = \text{diag}\{q_1, q_2, \dots, q_n\}$ so that the following LMIs (14) and (15) are satisfied:*

$$\Pi' = \begin{pmatrix} \Pi'_{11} & QB^T \\ * & -\epsilon I_n \end{pmatrix} < 0, \tag{14}$$

$$\epsilon L^T L - \beta Q < 0 \tag{15}$$

where $\Pi'_{11} = -QD - D^T Q - Y - Y^T + QAL + L^T A^T Q + \kappa Q$, $K = Q^{-1}Y$.

Remark 4 The model in Theorem 1 is FOMNNs, the memristive connection weights $\hat{a}_{ij}, \hat{a}_{ij}^\tau, \hat{b}_{ij}^\tau, \hat{b}_{ij}^\tau$ of (2) and (3) switch their values when their states change. If the connection values are not changed, the model will become traditional FNNs. Then, the synchronization criteria for FNNs can be obtained facilely.

Consider the following FOMNNs without time delays:

$$D^\alpha x(t) = -Dx(t) + (A + M\Lambda_1(t)E)f(x(t)) + I \tag{16}$$

$$D^\alpha y(t) = -Dy(t) + (A + M\Lambda_3(t)E)f(y(t)) + u(t) + I \tag{17}$$

Based on the controller (5), the error system is gotten as follows:

$$D^\alpha e(t) = -(D + K)e(t) + (A + \Delta A)g(e(t)) + \Sigma'(t) - \Gamma \text{sgn}(e(t)) \tag{18}$$

where $\Delta A = M\Lambda_3(t)E$, $\Sigma'(t) = [M(\Lambda_3(t) - \Lambda_1(t))E]f(x(t))$. Based on controller (5), the synchronization criteria for FOMNNs without time delays are obtained in Corollary 2.

Corollary 2 *Suppose that Assumptions 1 and 2 hold, under the proposed controller (5) with $\gamma_i = \sum_{j=1}^n (|\hat{a}_{ij} - \hat{a}_{ij}|)M_j$, the FOMNNs (16) and (17) can reach global asymptotical synchronization, if there exists a matrix Y , a positive constant ρ_1 and a positive definite matrix $Q = \text{diag}\{q_1, q_2, \dots, q_n\}$ so that the following LMI (19) is satisfied:*

$$\Pi'' = \begin{pmatrix} \Pi''_{11} & QM \\ * & -\rho_1 I_n \end{pmatrix} < 0, \tag{19}$$

where $\Pi''_{11} = -QD - D^T Q - Y - Y^T + QAL + L^T A^T Q + \rho_1 L^T E^T E L$, $K = Q^{-1}Y$.

Remark 5 If the communications among neural cells are without time delays, the model of drive-response FOMNNs will be reduced to (16) and (17). Then, the synchronization criteria for FMNNs with no delay can be obtained effortlessly.

4 Parameters Optimization by SIWPSO

The optimization model of the controller parameters is provided in this section. In order to get the optimal controller, the target function of the model is the sum of the control energy index and the ISE index. To achieve the synchronization of the FOMNNs (2) and (3), the criteria in Theorem 1 are chosen as the constraint conditions. However, the proposed optimization model is not easy to solve. The SIWPSO algorithm is applied to find out the optimal parameters.

4.1 The Optimization Model

The following target function is chosen to obtain the optimal parameters of the controller.

$$J = \int_0^\infty (u^T(t)u(t) + e^T(t)e(t))dt. \tag{20}$$

where $\int_0^\infty (u^T(t)u(t))dt$ is the control energy index, $\int_0^\infty (e^T(t)e(t))dt$ is the ISE index. Moreover, the synchronization conditions of the FOMNNs need to be achieved, which means that the proposed Theorem 1 needs to be reached. This suggests that for given constants $0 < \beta < \kappa$, there exists a matrix Y , some positive constants ϵ, ρ_1, ρ_2 and a positive definite matrix Q so that the LMIs (6) and (7) are satisfied:

$$\Pi = \begin{pmatrix} \Pi_{11} & QB^T & QM & 0 \\ * & -\epsilon I_n & 0 & Q\widehat{M} \\ * & * & -\rho_1 I_{n^2 \times n^2} & 0 \\ * & * & * & -\rho_2 I_{n^2 \times n^2} \end{pmatrix} < 0, \\ \epsilon L^T L - \beta Q < 0$$

where $\Pi_{11} = -QD - D^T Q - Y - Y^T + QAL + L^T A^T Q + \kappa Q + \rho_1 L^T E^T EL + \rho_2 \widehat{E}^T \widehat{E}$, $K = Q^{-1}Y$.

In brief, the parameter optimization of the controller is as follows:

$$\min J = \int_0^\infty (u^T(t)u(t) + e^T(t)e(t))dt, \\ s.t. \begin{cases} \Pi < 0, \\ \epsilon L^T L - \beta Q < 0, \\ Q > 0, \\ \epsilon > 0, \\ \rho_j > 0 (j = 1, 2). \end{cases} \tag{21}$$

where $\Pi_{11} = -QD - D^T Q - Y - Y^T + QAL + L^T A^T Q + \kappa Q + \rho_1 L^T E^T EL + \rho_2 \widehat{E}^T \widehat{E}$, $K = Q^{-1}Y$.

The parameter optimization of the controller is essentially to optimize the control gain matrix K . The optimal gain matrix can minimize the target function which includes the control energy index and ISE index. To solve the complicated calculation (21), the SIWPSO algorithm is put into use.

Remark 6 The target function (20) cannot be directly evaluated. In order to facilitate calculation, the Riemann sum definition of integral $\int_a^b F(x)dx \cong (b - a)F(c), c \in [a, b]$ is used to obtain an approximated value of the target function. The SIWPSO algorithm is an improved

intelligent algorithm to solve the optimization model (21). The better performance will be presented in simulation part.

4.2 SIWPSO for the Optimization Model

The SIWPSO algorithm requires each individual (particle) to maintain two vectors during evolution, namely the velocity vector $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iDm})$ and the position vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iDm})$, where i represents the number of particles, and Dm is the dimension of solutions. The velocity of a particle determines the direction and speed. The position reflects the position in the solution space. Both are the basis of evaluating the solution. The algorithm also requires each particle to maintain its own historical optimal position vector (represented by $\mathbf{pBest} = (pBest_i^1, pBest_i^2, \dots, pBest_i^{Dm})$) and a global optimal vector (represented by $\mathbf{gBest} = (gBest_i^1, gBest_i^2, \dots, gBest_i^{Dm})$), which guides particles to approach the global optimal region. The following are steps of the SIWPSO algorithm:

Step 1 Assign values to the parameters in the algorithm. For example, the particle swarm size m , maximum number of iterations N , and acceleration coefficient ac_1, ac_2 .

Step 2 Randomly initialize each particle.

Step 3 Evaluate each particle and get the global optimum.

Step 4 Evaluate the current solution with the consistent conditions (21). If they can synchronize the FOMNNs, go to the next step, whereas, go to the Step 7.

Step 5 Update (22) the velocity and position of every particle.

Step 6 Evaluate the fitness function (20) of each particle.

Step 7 Recalculate the historical optimum position $pBest_i$ of each particle.

Step 8 Recalculate the global optimal position $gBest_g$ of the group.

Step 9 Repeat steps 4-8 before reaching the maximum number of iterations.

Step 10 The optimal control gain is the latest $gBest_g$.

The update rules of position and velocity are as follows:

$$\begin{aligned} v_{id}(t + 1) &= \varpi(t)v_{id}(t) + ac_1\gamma_1(\mathbf{pBest}_{id}(t) - \mathbf{x}_{id}(t)) + ac_2\gamma_2(\mathbf{gBest}_{gd}(t) - \mathbf{x}_{id}(t)), \\ \mathbf{x}_{id}(t + 1) &= \mathbf{x}_{id}(t) + \mathbf{v}_{id}(t + 1), \end{aligned} \tag{22}$$

where i represents the i th particle and d is on behalf of the d th dimension in the search space. Usually, the inertia weight $\varpi(t)$ is chosen in (0.5, 1) randomly. Let $ac_1 = ac_2 = 2$. γ_1 and γ_2 are random numbers which are uniformly distributed at (0, 1).

Remark 7 The main difference between PSO and SIWPSO algorithm is the inertia weight. In PSO algorithm, the value of the inertia weight is usually set as 0.9, while in SIWPSO algorithm, it is randomly selected in (0.5, 1).

5 Simulation

A complete simulation is presented to exhibit the proposed theoretical approach and the optimization.

Consider the FOMNNs(1) with the following parameters: $\alpha = 0.98, \tau = 1, I = (0, 0)^T, f_i(x_i) = \tanh(x_i)(i = 1, 2), D = \text{diag}\{1, 1\}$,

$$A(x(t)) = \begin{pmatrix} a_{11}(x_1(t)) & a_{12}(x_1(t)) \\ a_{21}(x_2(t)) & a_{22}(x_2(t)) \end{pmatrix}, \quad B^\tau(x(t)) = \begin{pmatrix} b_{11}^\tau(x_1(t)) & b_{12}^\tau(x_1(t)) \\ b_{21}^\tau(x_2(t)) & b_{22}^\tau(x_2(t)) \end{pmatrix},$$

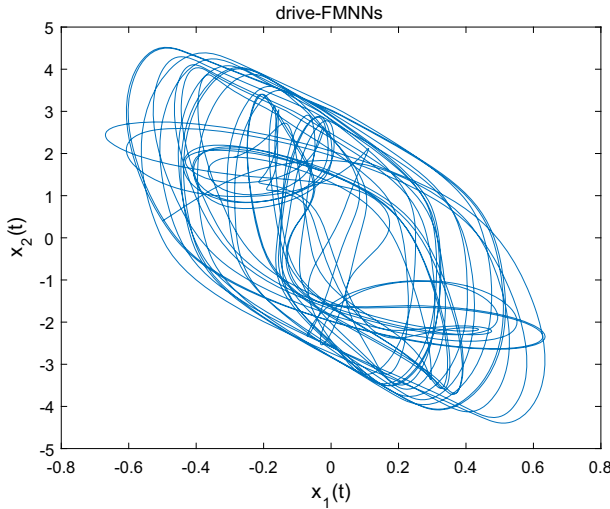


Fig. 1 Phase portrait of the drive-FOMNNs

where

$$\begin{aligned}
 a_{11}(x_1) &= \begin{cases} 1.75, & |x_1| \leq 3, \\ 2.1, & |x_1| > 3, \end{cases} & a_{12}(x_1) &= \begin{cases} -0.1, & |x_1| \leq 3, \\ -0.12, & |x_1| > 3, \end{cases} \\
 a_{21}(x_2) &= \begin{cases} -4.7, & |x_2| \leq 3, \\ -5.1, & |x_2| > 3, \end{cases} & a_{22}(x_2) &= \begin{cases} 2.85, & |x_2| \leq 3, \\ 3.2, & |x_2| > 3, \end{cases} \\
 b_{11}^\tau(x_1) &= \begin{cases} -1.7, & |x_1| \leq 3, \\ -1.6, & |x_1| > 3, \end{cases} & b_{12}^\tau(x_1) &= \begin{cases} -0.08, & |x_1| \leq 3, \\ -0.11, & |x_1| > 3, \end{cases} \\
 b_{21}^\tau(x_2) &= \begin{cases} -0.3, & |x_2| \leq 3, \\ -0.2, & |x_2| > 3, \end{cases} & b_{22}^\tau(x_2) &= \begin{cases} -2.5, & |x_2| \leq 3, \\ -2.38, & |x_2| > 3. \end{cases}
 \end{aligned}$$

It's easy to verify that $L_1 = L_2 = 1$, $M_1 = M_2 = 1$, $\gamma_1 = 0.5$, $\gamma_2 = 0.97$,

$$\begin{aligned}
 A &= \begin{pmatrix} 1.925 & -0.11 \\ -4.9 & 3.025 \end{pmatrix}, & B^\tau &= \begin{pmatrix} -1.65 & -0.095 \\ -0.25 & -2.44 \end{pmatrix}, \\
 M &= \begin{pmatrix} \sqrt{0.175} & \sqrt{0.01} & 0 & 0 \\ 0 & 0 & \sqrt{0.2} & \sqrt{0.175} \end{pmatrix}, & E &= \begin{pmatrix} \sqrt{0.175} & 0 & \sqrt{0.2} & 0 \\ 0 & \sqrt{0.01} & 0 & \sqrt{0.175} \end{pmatrix}^T, \\
 \hat{M} &= \begin{pmatrix} \sqrt{0.05} & \sqrt{0.015} & 0 & 0 \\ 0 & 0 & \sqrt{0.05} & \sqrt{0.06} \end{pmatrix}, & \hat{E} &= \begin{pmatrix} \sqrt{0.05} & 0 & \sqrt{0.05} & 0 \\ 0 & \sqrt{0.015} & 0 & \sqrt{0.06} \end{pmatrix}^T.
 \end{aligned}$$

The phase portrait of the drive-FOMNNs is shown in Fig. 1 with initial conditions $x = (-0.5, 0.4)^T$ for $t \in [-1, 0]$.

Take $\kappa = 1.5$, $\beta = 0.8$. By solving the LMIs (6) and (7) via the matlab toolbox, the feasible solutions are $\epsilon = 0.1494$, $\rho_1 = 1.1800$, $\rho_2 = 1.2099$, $Q = \text{diag}\{0.2541, 0.2541\}$,

$$Y = \begin{pmatrix} 1.6864 & -0.6069 \\ -0.6069 & 2.2883 \end{pmatrix}, \quad K = Q^{-1}Y = \begin{pmatrix} 6.6358 & -2.3883 \\ -2.3883 & 9.0042 \end{pmatrix}.$$

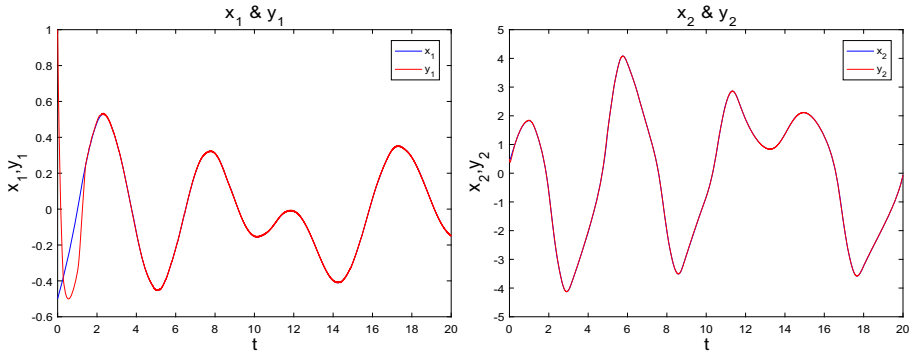


Fig. 2 State trajectories of the drive-response FOMNNs under the controller

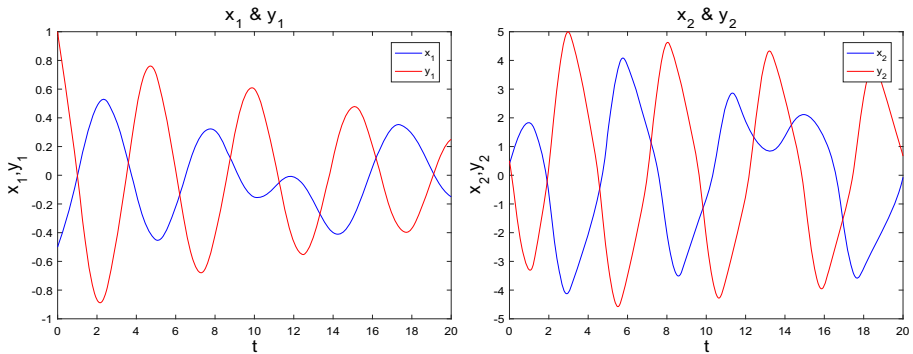


Fig. 3 State trajectories of the drive-response FOMNNs without controller

Based on Theorem 1, the driven-FOMNNs (2) can synchronize the response-FOMNNs (3) under the controller (5) for any initial values, which is verified by Figs. 2, 3 and 4. Figure 2 presents the state trajectories of the drive-response FOMNNs under the controller with initial values $x = (-0.5, 0.4)^T$, $y = (1, 0.5)^T$ for $t \in [-1, 0]$. Figure 3 presents the state trajectories of the drive-response FOMNNs without controller and the initial values are same with the Fig. 2. In Fig. 4, the left one describes the error trajectories of the FOMNNs with the controller and the right one without controller. It's obvious that the drive-response FOMNNs can be synchronized by the proposed controller. As time approaches infinity, the error system tends to zero.

The scale of the particle swarm is chosen as $m = 20$. The maximum iterations is chosen as $N = 30$. The unknown matrix

$$K = \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix},$$

the initial values of position are selected in $[-20, 20]$ randomly. The initial values of velocity are selected in $[-0.1, 0.1]$ randomly. After the performance of SIWPSO algorithm, the optimal solutions are presented in Table 1. As comparison groups, the optimal solutions by PSO and the solutions without optimizing process are presented in Table 1 as well. It's evident that the solutions searched by SIWPSO algorithm can realize the minimum of the

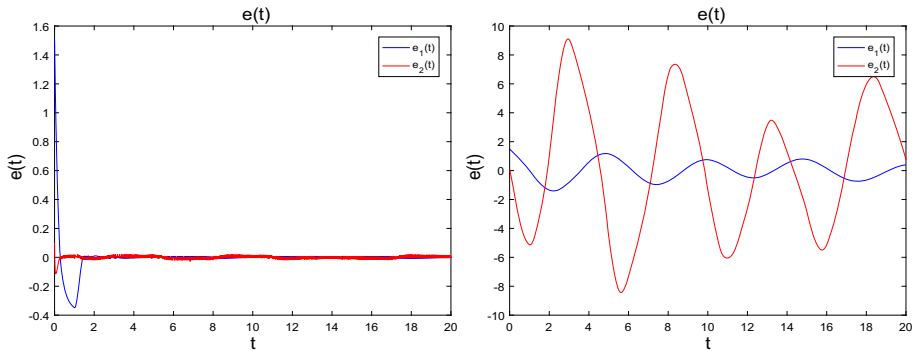


Fig. 4 Error trajectories with (left) and without (right) the controller

Table 1 Fitness function values for different values with K

Method	k_{11}	k_{12}	k_{21}	k_{22}	J value
LMI	6.6358	-2.3883	-2.3883	9.0042	6384.4
PSO	4.4182	-2.0266	-6.7339	16.2050	5567.9
SIWPSO	4.3560	-1.9227	-6.3219	16.8323	5552.3

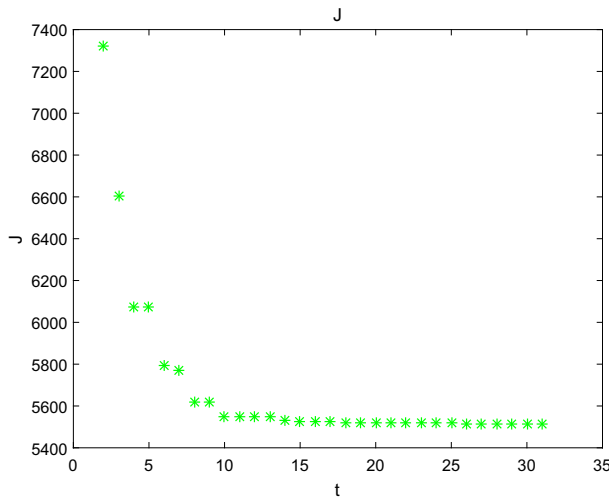


Fig. 5 The evolution of SIWPSO algorithm

fitness function . The solutions searched by PSO algorithm are also feasible, but they are not better than those searched by the SIWPSO algorithm.

Figure 5 illustrates the evolution process of the fitness function via SIWPSO algorithm. The conclusion can be drawn from the Fig. 5 that when the number of iterations increases, the value of J decreases from near 7400 to 5552.3, and at last, it does not change any more. In other word, the solutions which can minimize the value of J and let the value stable are the optimal control gains. The stable solution that minimizes the J value is the desirable optimization gain matrix of the controller.

6 Conclusion

In the paper, the synchronization of delayed memristive neural networks via optimal control using SIWPSO is studied. The networks are in fractional-order. First, a valid controller is designed, then based on some analytical methods, the drive-response synchronization results are received. Two corollaries without memristor and without time delays are provided at the same time. Second, this paper describes the optimization model of control parameters and the optimization process according to the SIWPSO algorithm. The optimization solutions can achieve the minimum target function, meanwhile, synchronize the drive-response FOMNNs. Finally, the proposed results can be confirmed by the simulations. Notably, a constant time delay is considered in this paper. If time delays are variant, maybe the delay partitioning approach can solve this problem. Besides, only the SIWPSO algorithm is considered in this paper. The genetic algorithm possibly has better effect, which will be studied in the future study.

Compliance with Ethical Standards

Conflict of interest The authors declare that there is no conflict of interest in preparing this article.

References

1. Namias V (1980) The fractional order Fourier transform and its application to quantum mechanics. *IMA J Appl Math* 25(3):241–265
2. Craiem D, Rojo FJ, Atienza JM, Armentano RL, Guinea GV (2008) Fractional-order viscoelasticity applied to describe uniaxial stress relaxation of human arteries. *Phys Med Biol* 53(17):4543
3. Azar AT, Vaidyanathan S, Ouannas A (2017) Fractional order control and synchronization of chaotic systems, vol 688. Springer, Berlin
4. Liu D, Zhang Y, Lou J, Lu J, Cao J (2018) Stability analysis of quaternion-valued neural networks: Decomposition and direct approaches. *IEEE Trans Neural Netw Learn Syst* 29(9):4201–4211
5. Zhang D, Kou KI, Liu Y, Cao J (2017) Decomposition approach to the stability of recurrent neural networks with asynchronous time delays in quaternion field. *Neural Netw* 94:55–66
6. Liu Y, Xu P, Lu J, Liang J (2016) Global stability of Clifford-valued recurrent neural networks with time delays. *Nonlinear Dyn* 2(84):767–777
7. Zhang J, Wu J, Bao H, Cao J (2018) Synchronization analysis of fractional-order three-neuron BAM neural networks with multiple time delays. *Appl Math Comput* 339:441–450
8. Bao H, Cao J, Kurths J, Alsaedi A, Ahmad B (2018) H_∞ state estimation of stochastic memristor-based neural networks with time-varying delays. *Neural Netw* 99:79–91
9. Bao H, Park JH, Cao J (2015) Exponential synchronization of coupled stochastic memristor-based neural networks with time-varying probabilistic delay coupling and impulsive delay. *IEEE Trans Neural Netw Learn Syst* 27(1):190–201
10. Zhang G, Shen Y, Wang L (2013) Global anti-synchronization of a class of chaotic memristive neural networks with time-varying delays. *Neural Netw* 46:1–8
11. Yang X, Cao J, Yu W (2014) Exponential synchronization of memristive Cohen–Grossberg neural networks with mixed delays. *Cogn Neurodyn* 8(3):239–249
12. Yang S, Guo Z, Wang J (2015) Robust synchronization of multiple memristive neural networks with uncertain parameters via nonlinear coupling. *IEEE Trans Syst Man Cybern Syst* 45(7):1077–1086
13. Bao H, Park JH, Cao J (2015) Adaptive synchronization of fractional-order memristor-based neural networks with time delay. *Nonlinear Dyn* 82(3):1343–1354
14. Wang F, Yang Y (2018) Intermittent synchronization of fractional order coupled nonlinear systems based on a new differential inequality. *Phys A* 512:142–152
15. Liu Y, Tong L, Lou J, Lu J, Cao J (2018) Sampled-data control for the synchronization of Boolean control networks. *IEEE Trans Cybern* 99:1–7

16. Wang F, Yang Y (2019) On leaderless consensus of fractional-order nonlinear multi-agent systems via event-triggered control. *Nonlinear Anal Model* 24(3):353–367
17. He W, Qian F, Lam J, Chen G, Han QL, Kurths J (2015) Quasi-synchronization of heterogeneous dynamic networks via distributed impulsive control: error estimation, optimization and design. *Automatica* 62:249–262
18. Yu W, Li C, Yu X, Wen G, Lv J (2018) Economic power dispatch in smart grids: a framework for distributed optimization and consensus dynamics. *China Inf Sci* 61(1):012204
19. Qin Q, Cheng S, Zhang Q, Li L, Shi Y (2016) Particle swarm optimization with interswarm interactive learning strategy. *IEEE Trans Cybern* 46(10):2238–2251
20. Perng JW, Chen GY, Hsu YW (2017) FOPID controller optimization based on SIWPSO-RBFNN algorithm for fractional-order time delay systems. *Soft Comput* 21(14):4005–4018
21. Chauhan P, Deep K, Pant M (2013) Novel inertia weight strategies for particle swarm optimization. *Memet Comput* 5(3):229–251
22. Huang Z, Cao J, Raffoul YN (2018) Hilger-type impulsive differential inequality and its application to impulsive synchronization of delayed complex networks on time scales. *China Inf Sci* 61:1–3
23. Chang W, Zhu S, Li J, Sun K (2018) Global Mittag–Leffler stabilization of fractional-order complex-valued memristive neural networks. *Appl Math Comput* 338:346–362
24. Chen J, Chen B, Zeng Z (2018) 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* 99:1–17
25. Perng JW, Chen GY, Hsieh SC (2014) Optimal PID controller design based on PSO-RBFNN for wind turbine systems. *Energies* 7(1):191–209
26. Fang H, Chen L, Shen Z (2011) Application of an improved PSO algorithm to optimal tuning of PID gains for water turbine governor. *Energy Convers Manag* 52(4):1763–1770
27. Chang Q, Yang Y, Sui X, Shi Z (2019) The optimal control synchronization of complex dynamical networks with time-varying delay using PSO. *Neurocomputing* 333:1–10
28. Kilbas AAA, Srivastava HM, Trujillo JJ (2006) *Theory and applications of fractional differential equations*, vol 204. Elsevier, Elsevier
29. Petrá I (2011) *Fractional-order nonlinear systems: modeling, analysis and simulation*. Springer, Berlin
30. Wang F, Yang Y (2018) Quasi-synchronization for fractional-order delayed dynamical networks with heterogeneous nodes. *Appl Math Comput* 339:1–14
31. Sanchez EN, Perez JP (1999) Input-to-state stability (ISS) analysis for dynamic neural networks. *IEEE Trans Circuits Syst I Fundam Theory Appl* 46(11):1395–1398
32. Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) *Linear matrix inequalities in system and control theory*, vol 15. SIAM, Philadelphia
33. Chen B, Chen J (2015) Razumikhin-type stability theorems for functional fractional-order differential systems and applications. *Appl Math Comput* 254:63–69
34. Chua L (2011) Resistance switching memories are memristors. *Appl Phys A* 102(4):765–783
35. Chen J, Zeng Z, Jiang P (2014) Global Mittag–Leffler stability and synchronization of memristor-based fractional-order neural networks. *Neural Netw* 51:1–8
36. Yang X, Ho DW (2016) Synchronization of delayed memristive neural networks: robust analysis approach. *IEEE Trans Cybern* 46(12):3377–3387
37. Bao H, Cao J, Kurths J (2018) State estimation of fractional-order delayed memristive neural networks. *Nonlinear Dyn* 94(2):1215–1225
38. Henderson J, Ouahab A (2009) Fractional functional differential inclusions with finite delay. *Nonlinear Anal Theory Methods Appl* 70(5):2091–2105
39. Gan Q (2013) Synchronization of competitive neural networks with different time scales and time-varying delay based on delay partitioning approach. *Int J Mach Learn Cybern* 4(4):327–337
40. Li K, Yu W, Ding Y (2015) Successive lag synchronization on nonlinear dynamical networks via linear feedback control. *Nonlinear Dyn* 80(1–2):421–430

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.