Research Article

Sliding Mode Matrix-Projective Synchronization for Fractional-Order Neural Networks

Jinman He,¹,² Tengfei Lei,³ and Limin Jiang

¹College of Science, Zhongyuan University of Technology, Zhengzhou 451191, China
²College of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450000, China
³School of Mechanical and Electrical Engineering, Qilu Institute of Technology, Jinan 250200, China

Correspondence should be addressed to Jinman He; hejinman1026@163.com and Tengfei Lei; leitengfeicanhe@126.com

Received 29 May 2021; Accepted 1 September 2021; Published 20 September 2021

Journal of Mathematics
Volume 2021, Article ID 4562392, 9 pages
https://doi.org/10.1155/2021/4562392

This work generalizes the projection scaling factor to a general constant matrix and proposes the matrix-projection synchronization (MPS) for fractional-order neural networks (FNNs) based on sliding mode control firstly. This kind of scaling factor is far more complex than the constant scaling factor, and it is highly variable and difficult to predict in the process of realizing the synchronization for the driving and response systems, which can ensure high security and strong confidentiality. Then, the fractional-order integral sliding surface and sliding mode controller for FNNs are designed. Furthermore, the criterion for realizing MPS is proved, and the reachability and stability of the synchronization error system are analyzed, so that the global MPS is realized for FNNs. Finally, a numerical application is given to demonstrate the feasibility of theory analysis. MPS is more general, so it is reduced to antisynchronization, complete synchronization, projective synchronization (PS), and modified PS when selecting different projective matrices. This work will enrich the synchronization theory of FNNs and provide a feasible method to study the MPS of other fractional-order dynamical models.

1. Introduction

Neural network is an important part of artificial intelligence, which is composed of a large number of highly connected neurons. It is a mathematical model based on the preliminary understanding of the physiological structure and activity mechanism of the brain. The neural network has the unique knowledge representation structure and can process information, learn, and adapt to the unknown system efficiently and quickly, which provide new research ideas for control problems and intelligent information processing. Integer order differential equations cannot describe the memory properties of neurons and the dependence on past history, but the fractional-order calculus [1–4], which has strong memory and hereditary characteristic, contains all of the information from the start point to the current moment and can describe the memory properties and dynamical behaviors of neurons more accurately. Therefore, FNNs can improve the computational ability of neurons, speed up the information transmission of neurons, and solve the problem of parameter identification effectively. With the development of fractional-order calculus, FNNs are becoming more and more popular and their dynamical behaviors have been widely investigated, such as stability [5–8], bifurcations [9], chaos and hyperchaos [10], and synchronization [11–14].

Two systems are called PS when the drive and response systems are synchronized to a scaling factor. Recently, scholars have researched the PS of FNNs and achieved a lot of valuable results. In [15–17], by using different control methods, the authors have researched the PS for FNNs. In [18], Zhang et al. have used the adaptive control method to achieve the PS of FNNs in quaternion field. In [19–21], researchers have explored the PS for fractional-order memristive neural networks with different characters. In [22–24], the authors extend the FNNs to the complex domain and study their PS and quasiprojective synchronization by using different control strategies. In [25], Ding and Shen constructed the fractional-order integral sliding mode
surface, designed fractional-order sliding mode controller, and realized PS for two FNNs with different structures. In [26, 27], by means of the fractional Lyapunov-like method, the authors realized PS in finite-time and mixed H∞/passive projective synchronization for nonidentical FNNs via a new sliding mode controller. In [28], by using sliding mode control, Wu et al. realized the finite-time interlayer PS of fractional-order two-layer networks based on Caputo derivatives.

To the best of our knowledge, the scaling factor of PS in most research studies is a diagonal matrix or a fixed constant, but in fact this kind of scaling factor may not ensure high security of communication. Our work will generalize the proportion factor to a general constant matrix, which is far more complex than the constant scaling factor. Also, it is highly variable and difficult to predict in the process of realizing the synchronization for the driving and response systems, which can ensure the high security and strong confidentiality. Based on the characteristics of matrix scaling factor, our work presents a new kind of MPS for FNNs, whose complexity and unpredictability can effectively increase the difficulty for hackers to track the right path, improve the antiattack capability of the system, and enhance the confidentiality of secure communication.

As everyone knows, synchronization of FNNs can be reached via various control methods. Especially, sliding mode control strategy has many important and special advantages, which include low sensitivity to the parameter perturbation and external disturbance, implementation simplicity, and fast response. In our work, because of the complexity of the MPS and nonidentical FNNs, it greatly increases the difficulty of control in actual operation. Hence, it is extremely necessary to use fractional-order sliding mode control strategy to research the global MPS.

According to the above discussion and main research content, our work is divided into the following chapters. Section 2 introduces some lemmas and establishes the FNNs. In Section 3, by using the fractional-order sliding mode control method, MPS is defined and sufficient criterion is proved. As applications, a numerical application is given to demonstrate theory analysis in Section 4. Section 5 concludes the whole work.

2. Preliminaries

**Definition 1** (see [1]). Fractional integral for function \( f : [t_0, +\infty) \rightarrow \mathbb{R} \)

\[
t_0^a t \, f(t) = \frac{1}{\Gamma(a)} \int_{t_0}^{t} (t - \xi)^{a - 1} f(\xi) d\xi, \quad (a > 0, t \geq t_0),
\]

where \( \Gamma(a) = \int_{0}^{\infty} e^{-t} t^{a-1} dt \) is gamma function.

**Definition 2** (see [1]). Caputo derivative for a function \( f(t) \in C^n([t_0, +\infty), R) \) is defined as

\[
t_0^a D^n_t f(t) = \frac{1}{\Gamma(n - a)} \int_{t_0}^{t} \frac{f^{(n)}(\xi)}{(t - \xi)^{a-n+1}} d\xi, \quad (a > 0, t \geq t_0),
\]

where \( n \) is the positive integer satisfying \( n - 1 < a < n \), \( f^{(n)} \) is the \( n \)-th order derivative of \( f(x) \), and \( C^n([t_0, +\infty), R) \) is the space which is composed of \( n \) order continuous differentiable functions from \([t_0, +\infty) \) to \( R \). In particular, when \( 0 < a < 1 \),

\[
t_0^a D^n_t f(t) = (1/\Gamma(1 - a)) \int_{t_0}^{t} (\xi/(t - \xi)^a) d\xi;
\]

when \( a = 1 \), the Caputo derivative operation \( t_0^a D^n_t f(t) \) is identified with integer order ones (\( df(t)/dt \)).

**Lemma 1.** If Caputo derivative \( t_0^a D^n_t f(t) \) is integrable, then

\[
t_0^a D^n_t f(t) = f(t) - \sum_{k=0}^{n-1} \frac{f^{(k)}(t_0)}{k!} (t - t_0)^k.
\]

Especially, for \( 0 < a < 1 \), one can obtain

\[
t_0^a D^n_t f(t) = f(t) - f(t_0).
\]

**Lemma 2.** For Caputo derivative \( t_0^a D^n_t f(t) \), the following equality holds:

\[
t_0^a D_1^a t_0^a D^n_t f(t) = t_0^a D_1^a t_0^a D^n_t f(t) - \frac{df(t)}{dt}.
\]

**Lemma 3.** If the constant \( C \neq 0 \), then \( t_0^a D^n_t C = 0 \).

**Lemma 4.** If function \( f(t) \in L^\infty([t_0, t], R) \), for \( a > 0 \) and \( \beta > 0 \), then

\[
t_0^a D^n t_0^a t_0^a D^n_1 f(t) = t_0^a D^n_1 D^n_1 f(t),
\]

when \( a = \beta \),

\[
t_0^a D^n_1 D^n_1 f(t) = f(t).
\]

**Lemma 5** (see [2]). For any time instant \( t \geq 0 \), if \( x(t) \in R \) is continuously differentiable, then the inequality \((1/2)^{n-\alpha} \int D_\alpha C^n t_0^a f(t)(x(t)) \leq x(t) \int D_\alpha C^n t_0^a x(t) \) holds. If \( x(t) \in R^n \), the inequality \((1/2)^{n-\alpha} \int D_\alpha C^n t_0^a x(t) \) also holds.

**Lemma 6** (see [6]). If \( f(t) \in C^1([0, +\infty), R) \) is a continuously differentiable function, the inequality

\[
t_0^a D^n_C f(t) \cdot (t^+)^{\alpha a} \leq \text{sgn}(f(t)) \int_0^t D^n_1 f(t)
\]

holds almost everywhere, where \( 0 < a < 1 \) and \( f(t^+) \leq \text{lim}_{t \to +}\), \( f(t) \).
Consider two non-identical FNNs as the drive and response system:
\[
\begin{align*}
\frac{C}{\alpha}D^\alpha_t x(t) &= -Cx(t) + Af(x(t)), \\
z(t) &= Px(t), \\
\frac{C}{\alpha}D^\alpha_t y(t) &= -Dy(t) +Bg(y(t)) + u(t), \\
\bar{z}(t) &= Py(t), \quad (0 < \alpha < 1),
\end{align*}
\] (8)

where \( x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T \in \mathbb{R}^n \) and \( y(t) = (y_1(t), y_2(t), \ldots, y_n(t))^T \in \mathbb{R}^n \) denote the state variables and \( z(t) \) and \( \bar{z}(t) \) are the outputs.

Assumption 1. The continuous neuron activation functions \( f \) and \( g \) meet Lipschitz condition
\[
|f(\phi) - f(\varphi)| < F|\phi - \varphi|, \\
|g(\phi) - g(\varphi)| < G|\phi - \varphi|, \quad (\phi, \varphi \in \mathbb{R}),
\] (10)

where \( F, G > 0 \) are Lipschitz constants.

\[\] 3. Main Results

In this part, based on the fractional sliding mode control method, the sliding mode controllers will be designed to research the MPS between systems (8) and (9).

The error function of MPS is
\[
e(t) = y(t) - \Lambda x(t),
\] (11)

where \( e = (e_1, e_2, \ldots, e_n)^T \) and \( \Lambda = (\Lambda_{ij})_{\text{non}} \) is a general constant matrix.

**Definition 3.** If any two solutions \( x(t) \) and \( y(t) \) with initial values \( x(0) \) and \( y(0) \) meet
\[
\lim_{t \to +\infty} \|e(t)\| = \lim_{t \to +\infty} \|y(t) - \Lambda x(t)\| = 0,
\] (12)

systems (8) and (9) are said to be MPS, where \( \Sigma_{i=1}^{\infty} \|y(t) - \Lambda x(t)\| \) denotes the Euclidean norm.

Taking Caputo derivative of both sides of error function \( e(t) = y(t) - \Lambda x(t) \) and substituting into (8) and (9), the error system can be obtained as

\[\] 3.1. Sliding Mode Controller Design. The design principle of sliding mode control is usually to design a suitable sliding surface as required and then construct a controller to force the system to move on the sliding surface and stay on it forever. First, the fractional integral sliding surface is designed as

\[
\frac{C}{\alpha}D^\alpha_t e(t) = \frac{C}{\alpha}D^\alpha_t y(t) - \Lambda \frac{C}{\alpha}D^\alpha_t x(t)
\]

\[\]

\[\]

\[\]

\[\]

where \( F, G > 0 \) are Lipschitz constants.

where \( F, G > 0 \) are Lipschitz constants.

**Definition 3.** If any two solutions \( x(t) \) and \( y(t) \) with initial values \( x(0) \) and \( y(0) \) meet

\[\] 3. Main Results

In this part, based on the fractional sliding mode control method, the sliding mode controllers will be designed to research the MPS between systems (8) and (9).

The error function of MPS is
\[
e(t) = y(t) - \Lambda x(t),
\] (11)

where \( e = (e_1, e_2, \ldots, e_n)^T \) and \( \Lambda = (\Lambda_{ij})_{\text{non}} \) is a general constant matrix.

**Definition 3.** If any two solutions \( x(t) \) and \( y(t) \) with initial values \( x(0) \) and \( y(0) \) meet
\[
\lim_{t \to +\infty} \|e(t)\| = \lim_{t \to +\infty} \|y(t) - \Lambda x(t)\| = 0,
\] (12)

systems (8) and (9) are said to be MPS, where \( \Sigma_{i=1}^{\infty} \|y(t) - \Lambda x(t)\| \) denotes the Euclidean norm.

Taking Caputo derivative of both sides of error function \( e(t) = y(t) - \Lambda x(t) \) and substituting into (8) and (9), the error system can be obtained as

\[\] 3.1. Sliding Mode Controller Design. The design principle of sliding mode control is usually to design a suitable sliding surface as required and then construct a controller to force the system to move on the sliding surface and stay on it forever. First, the fractional integral sliding surface is designed as

\[\]

\[\]

\[\]
then

\[ e(t) = e(0) + C_0^\alpha D_0^\alpha [-De(t) + B(g(y(t)) - g(\Lambda x(t))) - D\Lambda x(t) + Bg(\Lambda x(t)) - \Lambda (-Cx(t) + Af(x(t))) + u(t)]. \tag{17} \]

Next, from equations (15) and (17), we have

\[ S(t) = e(0) + C_0^\alpha I_0^\alpha [(C - D + KP)e(t) - D\Lambda x(t) + Bg(\Lambda x(t)) - \Lambda (-Cx(t) + Af(x(t))) + u(t)], \tag{18} \]

and sliding surface (18) and (15) are equivalent. Based on the sliding mode control method, when error system (13) moves on sliding surface, the formulas \( S(t) = 0 \) and \( \dot{S}(t) = 0 \) have to be satisfied. Then, one can obtain \( \dot{S}(t) = C_0^\alpha D_0^\alpha S(t) = 0 \) by using Lemma 2, which means \( C_0^\alpha D_0^\alpha S(t) = 0 \). Using (18) and Lemmas 3 and 4, we obtain

\[ C_0^\alpha D_0^\alpha S(t) = C_0^\alpha D_0^\alpha e(0) + C_0^\alpha D_0^\alpha I_0^\alpha [(C - D + KP)e(t) - D\Lambda x(t) + Bg(\Lambda x(t)) - \Lambda (-Cx(t) + Af(x(t))) + u(t)] \]

\[ = (C - D + KP)e(t) - D\Lambda x(t) + Bg(\Lambda x(t)) - \Lambda (-Cx(t) + Af(x(t))) + u(t) = 0. \tag{19} \]

From equation (19), the equivalent sliding mode controller is designed as

\[ u_{eq}(t) = -(C - D + KP)e(t) + D\Lambda x(t) - Bg(\Lambda x(t)) + \Lambda (-Cx(t) + Af(x(t))). \tag{20} \]

Substituting controller (20) into (13), the sliding mode error system is described as

\[ C_0^\alpha D_0^\alpha e(t) = -(C + KP)e(t) + B(g(y(t)) - g(\Lambda x(t))]. \tag{21} \]

\[ u(t) = u_{eq}(t) + u_r(t) \]

\[ = -(C - D + KP)e(t) + D\Lambda x(t) - Bg(\Lambda x(t)) + \Lambda (-Cx(t) + Af(x(t))) - k^* \text{sgn}(S(t)). \tag{23} \]

\[ V(t) = \frac{1}{2} S^T(t)S(t), \tag{24} \]

taking Caputo derivative of (24) with respect to time \( t \) and based on Lemma 5, we obtain

\[ C_0^\alpha D_0^\alpha V(t) \leq S^T(t)C_0^\alpha D_0^\alpha S(t) \]

\[ = S^T(t)[(C - D + KP)e(t) - D\Lambda x(t) + Bg(\Lambda x(t)) - \Lambda (-Cx(t) + Af(x(t))) + u(t)] \]

\[ = S^T(t)[-k^* \text{sgn}(S(t))] \]

\[ = -k^* \| S(t) \|. \tag{25} \]
As \( k^* > 0 \), system trajectories asymptotically converge to \( S(t) = 0 \), which means synchronization error trajectories \( (21) \) reach the predetermined sliding surface globally and stay on it forever. \( \square \)

3.2. Stability Analysis

**Theorem 2.** If Assumption 1 and the inequality

\[
C_i > \sum_{j=1}^{n} \left( \sum_{l=1}^{n} [k_{ij} P_{il}] \right) + \sum_{j=1}^{n} [b_{ij}] m_j, \tag{26}
\]

hold, then the drive system (8) and response system (9) can realize MPS on controller (23).

Proof. First, error system (21) is converted as

\[
C_0^D_t e_i(t) = -C_i e_i(t) - \sum_{j=1}^{n} \left( \sum_{l=1}^{n} k_{ij} P_{lj} \right) e_j(t) + \sum_{j=1}^{n} b_{ij} \left( g_j \left( y_j(t) - g_j \left( \sum_{i=1}^{n} \Delta_{ji} x_i(t) \right) \right) \right).
\]

Next, design Lyapunov function \( V(t) = \| e(t) \|_1 = \sum_{i=1}^{n} |e_i(t)| \), according to Assumption 1 and Lemma 6, taking Caputo derivative of Lyapunov function \( V(t) \) along error trajectories (21), then

\[
C_0^D_t V(t) = \sum_{i=1}^{n} \sum_{j=1}^{n} [b_{ij}] G_j [e_j(t)].
\]

Let \( \lambda_i = C_i - \sum_{j=1}^{n} \left( \sum_{l=1}^{n} [k_{ij} P_{lj}] \right) - \sum_{j=1}^{n} [b_{ij}] G_j \), and we can obtain

\[
\lambda = \min_{1 \leq i \leq n} \lambda_i \leq \min_{1 \leq i \leq n} \left( C_i - \sum_{j=1}^{n} \left( \sum_{l=1}^{n} [k_{ij} P_{lj}] \right) - \sum_{j=1}^{n} [b_{ij}] G_i \right) > 0,
\]

then

\[
C_0^D_t V(t) \leq -\lambda V(t) = -\lambda \| e(t) \|_1 \leq 0, \quad (t \geq 0).
\]

and \( V(t) \) is monotonously nonincreasing. Hence, \( V(t) \leq V(0)(t \geq 0) \) and \( e_i(t) \) is bounded on \( t \geq 0 \) from the definition of \( V(t) \). There exists a positive constant satisfying \( |C_0^D_t e(t)||e(t)||_1| \leq M_1 \); then, we claim that \( \lim_{t \to \infty} \| e(t) \|_1 = 0 \) whose proof is similar to Theorem 1 in [16]. Therefore, error system (21) is globally asymptotically stable and MPS between systems (8) and (9) is realized based on controller (23). \( \square \)

**Remark 1.** When activating controller (23), response system (9) converts to

\[
C_0^D_t y(t) = -(C + KP)y(t) + (CA - \Delta C)x(t) + kp\Delta x(t) + Bg(y(t)) - Bg(\Delta x(t)) + AAf(x(t)),
\]

and then based on drive system (8), controlled response system (31), and error system (27), the MPS behaviors between systems (8) and (9) can be analyzed furthermore.

**Remark 2.** Theorems 1 and 2 are still true and new for \( \alpha = 1 \).

In particular, the MPS is a kind of more general synchronization. Selecting different projection matrices and controllers, it degenerates to some special synchronization types as Remark 3.
Remark 3

(1) If $\Lambda = I$, the sliding mode controller becomes
$$u(t) = u_{eq}(t) + u_r(t)$$
$$= -(C - D + KP)y(t) + KP\dot{x}(t) - Bg(x(t)) + Af(x(t)) - k^*\text{sgn}(S(t)),$$
where sliding surface $S(t)$ is given by (15), and then systems (8) and (9) can achieve complete synchronization.

(2) If $\Lambda = -I$, based on sliding surface (15), the sliding mode controller becomes
$$u(t) = u_{eq}(t) + u_r(t)$$
$$= -(C - D + KP)y(t) - Bg(x(t)) - k^*\text{sgn}(S(t)),$$
and then systems (8) and (9) can realize antisynchronization.

(3) If $\Lambda = cI$ ($c$ = const and $c \neq \pm 1$), the sliding mode controller is similar to (23) and sliding surface $S(t)$ is given by (15), and then systems (8) and (9) can achieve the PS.

(4) If $\Lambda = \text{diag}(c_1, c_2, \ldots, c_n)$ ($c_i$ = const, $i = 1, 2, \ldots, n$) and at least two $c_i$ of them are unequal, the sliding mode controller is similar to (23) and sliding surface $S(t)$ is given by (15), and then systems (8) and (9) can achieve the modified PS.

4. Numerical Application

Two nonidentical drive and response systems are considered as
Figure 2: MPS behaviors between systems (34) and (35). Time history of (a) $x_1, y_1$; (b) $x_2, y_2$; (c) $x_3, y_3$; (d) $x_1 + x_2 - 4x_3, y_1$; (e) $-0.5x_1 - 2x_2 + x_3, y_2$; and (f) $-3x_1 - 0.5x_2 - x_3, y_3$. 
and then error functions are calculated as

$$\begin{align*}
e_1 &= y_1 - \sum_{j=1}^{3} A_1 j x_j = y_1 - (x_1 + x_2 - 4x_3), \\
e_2 &= y_2 - \sum_{j=1}^{3} A_2 j x_j = y_2 - (-0.5x_1 - 2x_2 + x_3), \\
e_3 &= y_3 - \sum_{j=1}^{3} A_3 j x_j = y_3 - (-3x_1 - 0.5x_2 - x_3).
\end{align*}$$

Next, choosing $k_1 = 5, k_2 = 5, k_3 = 5,$ and $k^* = 20$ and according to Theorems 1 and 2, the MPS between systems (34) and (35) is realized.

Figures 2(a)–2(c) depict the time trajectories of variables $x_1 \sim y_1, x_2 \sim y_2,$ and $x_3 \sim y_3$. Simultaneously, it is clearly seen that variables $x_1 + x_2 - 4x_3 \sim y_1, -0.5x_1 - 2x_2 + x_3 \sim y_2,$ and $-3x_1 - 0.5x_2 - x_3 \sim y_3$ realize the complete synchronization as depicted in Figures 2(d)–2(f). Figure 3 describes that the synchronization error trajectories converge to zero asymptotically for drive system (34) and response system (35). These numerical simulations and figures clearly indicate the applicability and effectiveness of the sliding controller for MPS.

5. Conclusions

For general FNNs, the paper comes up with the global MPS, constructs the fractional-order sliding surface, designs a sliding mode controller, establishes and proves the sufficient condition, and then realizes the global MPS. Our theory analysis provides important theoretical basis and technical support for enhancing signal security by using MPS of FNNs and contributes to the development of artificial intelligence. In future works, we will extend the MPS to the fractional-order memristive neural networks and consider applying them into secret communication.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

This work was supported by the National Natural Science Foundation of China (12102492).

References

[1] A. K. Anatoly, *Theory and Applications of Fractional Differential Equations*, Elsevier, Netherlands, 2006.
[2] N. Aguila-Camacho, M. A. Duarte-Mermoud, and J. A. Gallegos, “Lyapunov functions for fractional order systems,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 9, pp. 2951–2957, 2014.
[3] L. Wu, S. Liu, and Y. Yang, “Using the fractional order method to generalize strengthening buffer operator and weakening buffer operator,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 6, pp. 1074–1078, 2018.
[4] Y. Chen, L. F. Wu, L. Y. Liu, and K. Zhang, “Fractional Hausdorff grey model and its properties,” *Chaos, Solitons & Fractals*, vol. 138, Article ID 109915, 2020.
[5] S. Liang, R. Wu, and L. Chen, “Comparison principles and stability of nonlinear fractional-order cellular neural networks with multiple time delays,” *Neurocomputing*, vol. 168, pp. 618–625, 2015.
[6] S. Zhang, Y. Yu, and H. Wang, “Mittag-Leffler stability of fractional-order Hopfield neural networks,” *Nonlinear Analysis: Hybrid Systems*, vol. 16, pp. 104–121, 2015.
[7] L. Zhang, Q. Song, and Z. Zhao, “Stability analysis of fractional-order complex-valued neural networks with both leakage and discrete delays,” *Applied Mathematics and Computation*, vol. 298, pp. 296–309, 2017.
[8] L.-L. Huang, J. H. Park, G.-C. Wu, and Z.-W. Mo, “Variable-order fractional discrete-time recurrent neural networks,” *Journal of Computational and Applied Mathematics*, vol. 370, Article ID 112633, 2020.
[9] C. Huang, J. Cao, M. Xiao, A. Alsaedi, and T. Hayat, “Bifurcations in a delayed fractional complex-valued neural network,” *Applied Mathematics and Computation*, vol. 292, pp. 210–227, 2017.
[10] X. Huang, Z. Zhao, Z. Wang, and Y. Li, “Chaos and hyperchaos in fractional-order cellular neural networks,” *Neurocomputing*, vol. 94, pp. 14–21, 2012.
[11] H.-L. Li, L. Zhang, C. Hu, H. Jiang, and J. Cao, “Global Mittag-Leffler synchronization of fractional-order delayed quaternion-valued neural networks: direct quaternion approach,” *Applied Mathematics and Computation*, vol. 373, Article ID 125020, 2020.
[12] J. He, F. Chen, T. Lei, and Q. Bi, “Global adaptive matrix-projective synchronization of delayed fractional-order competitive neural network with different time scales,” *Neural Computing & Applications*, vol. 32, no. 16, pp. 12813–12826, 2020.
[13] X. Huang, Y. Fan, J. Jia, Z. Wang, and Y. Li, “Quasi-synchronisation of fractional-order memristor-based neural networks with parameter mismatches,” *IET Control Theory & Applications*, vol. 11, no. 14, pp. 2317–2327, 2017.
[14] J. M. He, F. Q. Chen, and Q. S. Bi, “Quasi-matrix and quasi-inverse-matrix projective synchronization for delayed and disturbed fractional order neural network,” *Complexity*, vol. 2019, Article ID 4823709, 15 pages, 2019.
[15] H. Q. Wu, L. F. Wang, Y. Wang, P. F. Niu, and B. L. Fang, “Global Mittag-Leffler projective synchronization for fractional-order neural networks: an LMI-based approach,” *Advances in Difference Equations*, vol. 132, 2016.