Journal of Computational Mathematics
Vol.44, No.1, 2026, 213–231.

http://www.global-sci.org/jcm
doi:10.4208/jcm.2410-m2024-0025

CHAOTIC SYNCHRONIZATION OF QUATERNIONIC NEURAL NETWORKS WITH TIME-VARYING DELAYS AND ITS APPLICATION IN CRYPTOSYSTEM*

Li Li[1)]
College of Electrical and Control Engineering, College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an 710054, China;
Xi'an Key Laboratory of Electrical Equipment Condition Monitoring and Power Supply Security, Xi'an 710054, China
Email: lilxiansen@163.com
Xudong Chen and Jing Liang
College of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China
Emails: cxd2429771532@163.com, Liang607997@163.com
Farong Kou
College of Electrical and Control Engineering, College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an 710054, China
Email: koufarong@xust.edu.cn
Hongguang Pan
College of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China;
Xi'an Key Laboratory of Electrical Equipment Condition Monitoring and Power Supply Security, Xi'an 710054, China
Email: hongguangpan@163.com

Abstract

For complex-valued or quaternionic neural networks, scholars and researchers usually decompose them into real-valued systems. The decomposed real-valued systems are equivalent to original systems. Then, the dynamical behaviors of real-valued systems obtained are investigated, including stability, synchronization, and chaos etc. In this paper, a class of quaternionic neural networks with time-varying delays is investigated. First, by designing a suitable PI controller, synchronization of the considered chaotic system is realized. By using a non-decomposition method and structuring a novel Lyapunov functional, sufficient conditions are derived to guarantee synchronization between the drive-response systems. It is worth mentioning that, unlike other methods, our approach does not require breaking down the quaternionic neural networks into four separate real-valued systems. Furthermore, we demonstrate the practical application of these chaotic quaternionic neural networks with time-varying delays in image encryption and decryption. Based on one sequence of chaotic signal from state trajectory of single quaternion-valued neuron and a new encryption algorithm, the application of chaotic system proposed, that is, image encryption, is researched. The process of image decryption is simply the reverse of the encryption process. Finally, numerical simulation examples are provided to validate the effectiveness of the designed PI controller and performance of image encryption and decryption.

* Received April 18, 2024 / Revised version received June 6, 2024 / Accepted October 14, 2024 / Published online November 18, 2024 /
1) Corresponding author

Mathematics subject classification: 92B20.
Key words: Quaternionic, Neural network, Chaotic synchronization, Image encryption and decryption, Time-varying delays.

1. Introduction

Quaternions, as a type of hypercomplex, have been successfully applied in many fields, such as computer vision, crystallographic texture and quantum mechanics [11,17,47], etc. Quaternion can efficiently reduce the computational complexity of high-dimensional problems [6]. In view of the advantages of quaternion, some scholars and researchers propose and model quaternionic neural networks (QNNs). QNNs have significant applications in color image compression, color night vision, robots and other fields [31, 33, 40, 48, 54]. Compared with complex-valued neural networks (CVNNs), QNNs with higher storage capacity can directly deal with 3D and 4D problems which can also be solved with a great number of neurons in CVNNs [8,10,21,22,34,48]. So far, researches on the dynamics of QNNs have made a lot of achievements [9,12,15,43], just to cite a few. Chen *et al.* [9] investigated the problem of stability for continuous-time and discrete-time QNNs. In [15, 24], stability for a delayed fractional-order QNNs was researched. The state estimation problem of QNN with time-varying delays (TVDs) was studied by extending Jensen's inequality to the quaternion domain [43]. In [12], authors studied the stability problem of a class of stochastic QNNs with TVDs using stochastic analysis techniques and Lyapunov functions (LF). In [51], the Mittag-Leffler stability and synchronization of Caputo fractional-order fuzzy QNNs with proportional delay and derivative order intervals were studied. In [20], by adopting an extended modification Lyapunov-Razumikhin method, global finite-time stability problem for QNNs was researched. In [23], exponential estimation and passivity of memristive neural networks (NNs) with quaternion parameters were studied. A Takagi-Sugeno type rule was introduced into the quaternion memristive NNs, which makes the system much easier. In [39], by using non-separation method, the global asymptotic stability problem for Takagi-Sugeno fuzzy quaternion-valued bidirectional associative memory NNs with discrete, distributed and leakage delays was investigated.

Proverbially, the signal transmission between neurons inevitably exists time delay [36, 42], which affect the dynamical behaviors of NNs, including chaos, oscillation, instability, etc. [7, 30, 35, 37, 38, 50]. For chaotic NNs, it has the characteristics of noise-like behavior and non-periodicity, and follows deterministic rules. Due to chaotic NNs are sensitive to initial conditions [49], small changes in initial conditions will have a significant impact on signal behavior. These characteristics make chaotic NNs good candidates for cryptosystem [19, 45]. At present, many results about chaotic NNs have obtained [1, 14, 32, 52, 53], just to cite a few. The asymptotic stability of transiently chaotic NNs is considered in [52]. Through observer based sliding mode control, Zhao *et al.* [53] studied the synchronization problem of delayed chaotic NNs with unknown disturbances. A secure communication scheme based on event triggered strategy and master-slave NNs quantization synchronization was proposed in [14]. The memristive chaotic NNs and the cascaded chaotic NNs were presented in [32] and [1], respectively.

With the increase in communication frequency on open networks, secure communication is becoming increasingly important. A good encryption process should be sensitive to cipher key, and the key space should be large enough to make brute-force attacks infeasible [2]. Since chaotic NNs have both the characteristic of NNs and chaos, chaotic NNs are applied to secure communication [3, 25]. Thus, the researches on dynamical properties of delayed chaotic NNs

have theoretical meaning and actual value [46]. The application of chaos synchronization in image secure communication mainly includes chaotic shift keying, chaotic modulation and chaos masking. Based on chaos master-slave network synchronization, chaos masking has been widely used in image encryption [4, 13, 16, 18, 27, 29, 44].

Based on the above discussions and considering the advantages of chaotic systems and QNNs with TVDs, we study the synchronization of chaotic QNNs with TVDs and its application in cryptosystems. In most cases, confusion is not desirable and needs to be controlled. On the one hand, by designing an appropriate PI controller, the synchronous control of the drive-response chaotic systems is studied, and the responsive chaotic system drives the chaotic system to make the chaotic trajectory of the two systems consistent. On the other hand, the application of chaotic QNNs with TVDs is adopted in image encryption and decryption. The main contributions of this paper are as follows. In most cases, chaos is undesirable and needs to be controlled. On one hand, by designing a suitable PI controller, the synchronization control of the drive-response chaotic systems is studied. On the other hand, the application of chaotic QNNs with TVDs is adopted in image encryption and decryption. The main contributions of this paper are as follows:

• For complex-valued or QNNs, scholars and researchers usually decompose them into real-valued systems. The dynamical behaviors of real-valued systems obtained are investigated, including stability, synchronization, and chaos etc. In this paper, by using non-decomposition methods, we construct a novel LF in form of quaternion self-conjugate matrices, and sufficient conditions are derived to guarantee the synchronization of the drive-response systems. Here, QNNs do not be decomposed into four real-valued systems.

• Chaotic systems are characterized by noise-like behaviors and non-periodical, and it follows deterministic rules. In most cases, chaos is undesirable and needs to be controlled. Thus, it is of great importance to investigate synchronization problems of chaotic systems. By designing a suitable PI controller, synchronization of the proposed system is realized in this paper. Through non-decomposition method and LF in form of quaternion self-conjugate matrices, sufficient conditions are derived to guarantee synchronization of drive-response systems.

• In this paper, based on one sequence of chaotic signal from state trajectory of single quaternionic neuron and a new encryption algorithm, the application of chaotic system proposed, that is, image encryption and decryption, is researched. Using chaotic sequences and designed encryption algorithms, the confidentiality ability of image data transmission communication will be improved.

The structure and organization of this paper is as this: the QNNs model considered is presented, and some definitions, lemmas and assumptions are given in Section 2. In Section 3, the chaotic synchronization of the studied NNs is researched. Some criteria are derived to guarantee the synchronization of two QNNs with TVDs. Results in Section 3 are illustrated by numerical examples in Section 4. Besides, for chaotic QNNs, its application in image encryption and decryption is also presented. Finally, Section 5 concludes the paper.

Notations: $\mathbb{R}$ is the set of real numbers. $\mathbb{Q}$ and $\mathbb{Q}^n$ denote the set of quaternions and n-dimensional quaternionic vectors. $\mathbb{Q}^{n\times n}$ denotes $n \times n$ matric with entire from $\mathbb{Q}$. $\overline{A}$ denotes the conjugate of matrix $A \in \mathbb{A}^{n\times m}$. A^H stands for the conjugate transpose of matrix $A \in \mathbb{Q}^{n\times m}$. The notation $\star$ denotes the Hermitian symmetry (or conjugate transpose) of a suitable block in a Hermitian matrix. $\mathbb{C}([-\tau, 0], \mathbb{Q}^n)$ denotes the set of all continuous functions from $[-\tau, 0]$ to $\mathbb{Q}^n$.

2. Preliminaries and Model Description

In this section, some definitions, assumptions and lemmas are given.

Definition 2.1. *The quaternion-valued activation function $f(x)$ is defined as*

$$f(x) = f^R(x^R) + \boldsymbol{i} f^I(x^I) + \boldsymbol{j} f^J(x^J) + \boldsymbol{k} f^K(x^K),$$

where $x = x^R + \boldsymbol{i}x^I + \boldsymbol{j}x^J + \boldsymbol{k}x^K \in \mathbb{Q}$, $x^R, x^I, x^J, x^K \in \mathbb{R}$.

Definition 2.2 (Arnold Cat Map Permutation Algorithm, [5]). *Suppose the size of original gray-scale image M is $N \times N$ and the pixel positions coordinates are*

$$S = \{(x, y) \,|\, x, y = 1, 2, \ldots, N\}.$$

Arnold cat map is described as follows:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \Gamma\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = A \begin{bmatrix} x \\ y \end{bmatrix} \bmod(N) = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod(N),$$

where mod *denotes the modulo of $A\begin{bmatrix} x \\ y \end{bmatrix}$ and N, p and q are called the cat map control parameters, which are positive integers, (x, y) is original position, $(\hat{x}, \hat{y})$ is new positions.*

Remark 2.1. If $\det(A) = 1$ holds, then the cat map is area-preserving. Moreover, after several iterations, the original image M can be permuted completely.

Throughout this paper, the following assumptions on the activation function $f(\cdot)$ should be needed.

Assumption 2.1. For any $j \in \{1, 2, \ldots, n\}$, there exists a constant $\lambda_i \in \mathbb{R}$ such that

$$|f_j(p) - f_j(q)| \leq \lambda_j |p - q|, \quad p, q \in \mathbb{Q}.$$

For convenience, one denotes $L = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$.

Lemma 2.1. *For any $x, y \in \mathbb{Q}$,*
(1) $\overline{\overline{x}} = x$; (2) $x + \overline{x} = 2x^R$; (3) $\overline{xy} = \overline{yx}$.

Lemma 2.2 ([7]). *Let $A, B \in \mathbb{Q}^{n\times n}$, $p, q \in \mathbb{Q}$. Then*

(1) $\boldsymbol{j}C = \overline{C}\boldsymbol{j}$ or $\boldsymbol{j}C\boldsymbol{j}^H = \overline{C}$ for any complex matric $C \in \mathbb{C}^{n\times n}$;

(2) $(A^H)^{-1} = (A^{-1})^H$ if A is invertible;

(3) $(AB)^H = B^H A^H$;

(4) every quaternion q can be uniquely expressed as $q = q_1 + q_2\boldsymbol{j}$, where $q_1, q_2 \in \mathbb{C}$;

(5) $|p + q| \leq |p| + |q|$ and $|pq| = |p||q|$;

(6) $(AB)^{-1} = B^{-1}A^{-1}$, if A, B are invertible.

Lemma 2.3 ([8]). *Let $R\in\mathbb{Q}^{n\times n}$ be a Hermitian matrix and $R=R_1+R_2\boldsymbol{j}$, where $R_1, R_2\in\mathbb{C}^{n\times n}$. Then, $R<0$ is equivalent to*

$$\begin{pmatrix} R_1 & -R_2 \\ \overline{R}_2 & \overline{R}_1 \end{pmatrix} < 0.$$

Lemma 2.4 ([8]). *Let $R\in\mathbb{Q}^{n\times n}$ be a positive definite Hermitian matrix, and $\phi:[a,b]\to\mathbb{Q}^n$ be a vector-valued function of one real variable, where $[a,b]\subset\mathbb{R}$. Suppose the integrations concerned are well defined. Then*

$$(b-a)\int_a^b \phi^H(s)R\phi(s)ds \le \Big(\int_a^b \phi(s)ds\Big)^H R\Big(\int_a^b \phi(s)ds\Big).$$

The drive chaotic system is a kind of QNNs with TVDs, which is described as

$$\dot{x}_i(t) = -d_i x_i(t) + \sum_{j=1}^{n} b_{ij} f_j\big(x_j(t)\big) + \sum_{j=1}^{n} c_{ij} f_j\big(x_j\big(t-\tau_j(t)\big)\big) + I_i(t), \tag{2.1}$$

where $x_i(t)$ denotes the quaternion-valued state of the i-th neuron at time t. $f_j(x_j(t))\in\mathbb{Q}$ denotes the output of the j-th neuron, as defined in Definition 2.1. $\tau_j(t)$ is the transmission delay with $0\le\tau_j(t)\le\tau$ and $\dot{\tau}(t)\le\mu$. $I_i(t)\in\mathbb{Q}$ is external input, b_{ij} and c_{ij} are quaternions, b_{ij} and c_{ij} denote the feedback weights, $d_i>0$ is the damping gain.

The initial condition of (2.1) is given as

$$x_k(s) = \phi_k(s), \quad s\in[-\tau,0],$$

where

$$\phi_k(s) = \phi_k^R(s) + \boldsymbol{i}\phi_k^I(s) + \boldsymbol{j}\phi_k^J(s) + \boldsymbol{k}\phi_k^K(s), \quad k=1,2,\ldots,n, \quad \phi_k(s)\in\mathbb{C}([-\tau,0],\mathbb{Q}^n).$$

An equivalent vector form of drive system (2.1) is

$$\dot{x}(t) = -Dx(t) + Bf\big(x(t)\big) + Cf\big(x\big(t-\tau(t)\big)\big) + I, \tag{2.2}$$

where

$$\begin{aligned}
&x(t) = \big(x_1(t), x_2(t), \cdots, x_n(t)\big)^\top \in \mathbb{Q}^n,\\
&D = \operatorname{diag}\{d_1, d_2, \cdots, d_n\} \in \mathbb{R}^{n\times n}, \quad d_i>0, \quad i=1,2,\ldots,n,\\
&B = (b_{ij})_{n\times n} \in \mathbb{Q}^{n\times n},\\
&C = (c_{ij})_{n\times n} \in \mathbb{Q}^{n\times n},\\
&I = \big(I_1(t), I_2(t), \cdots, I_n(t)\big)^\top \in \mathbb{Q}^n,\\
&f\big(x(t)\big) = \big(f_1\big(x_1(t)\big), f_2\big(x_2(t)\big), \cdots, f_n\big(x_n(t)\big)\big)^\top \in \mathbb{Q}^n,\\
&f\big(x\big(t-\tau(t)\big)\big) = \big(f_1\big(x_1\big(t-\tau_1(t)\big)\big), x_2\big(t-\tau_2(t)\big), \cdots, x_n\big(t-\tau_n(t)\big)\big)^\top \in \mathbb{Q}^n.
\end{aligned}$$

The response chaotic system is given as follows:

$$\dot{y}_i(t) = -d_i y_i(t) + \sum_{j=1}^{n} b_{ij} f_j\big(y_j(t)\big) + \sum_{j=1}^{n} c_{ij} f_j\big(y_j\big(t-\tau_j(t)\big)\big) + I_i(t) + u_i(t), \tag{2.3}$$

where $u_i(t)$ is the controller and will be designed later. An equivalent vector form of response system (2.3) is

$$\dot{y}(t) = -Dy(t) + Bf\big(y(t)\big) + Cf\big(y\big(t-\tau(t)\big)\big) + I + U, \tag{2.4}$$

where

$$U = \big(u_1(t), u_2(t), \cdots, u_n(t)\big)^\top \in \mathbb{Q}^n.$$

Define error $e(t) = y(t) - x(t) \in \mathbb{Q}^n$. The error system is shown in the following equation:

$$\dot{e}(t) = -De(t) + B\hat{f}\big(e(t)\big) + C\hat{f}\big(e\big(t-\tau(t)\big)\big) + U, \tag{2.5}$$

where

$$\begin{aligned}
&\hat{f}\big(e(t)\big) = f\big(y(t)\big) - f\big(x(t)\big),\\
&\hat{f}\big(e\big(t-\tau(t)\big)\big) = f\big(y\big(t-\tau(t)\big)\big) - f\big(x\big(t-\tau(t)\big)\big),\\
&e(t) = \big(e_1(t), e_2(t), \cdots, e_n(t)\big)^\top \in \mathbb{Q}^n.
\end{aligned}$$

Remark 2.2. The purpose of this paper is to ensure drive chaotic systems (2.2) synchronizes with response chaotic systems (2.4) based on the proposed controller. By designing two adaptive gain matrices, global asymptotic stability of error system (2.5) will be guaranteed. In addition, the application of chaotic systems, that is, image encryption and decryption, will be presented in Example 4.2.

Remark 2.3. Quaternion differential equations are a special form of differential equations, which are widely used in physics, engineering, computer graphics and other fields. Unlike ordinary differential equations, the variable in a quaternion differential equation is a quaternion, i.e. an extended complex system composed of real and imaginary parts. The quaternionic differential equation can represent the state of the physical system in a four-dimensional space over time. Therefore, it is widely used in the fields of robot control, motion planning, and spatial pose control. Moreover, in computer graphics, quaternionic differential equations are used to model the transformations of object rotation, scaling, twist and etc. In conclusion, quaternion differential equations are a mathematical tool with wide applications that can be used to describe many physical and engineering problems. There are many solutions, but the Euler-Rhode formula is one of the most commonly used methods. In-depth study of quaternionic differential equations is of great significance for a deep understanding of related issues in the fields of physics, engineering and computer graphics.

3. Synchronization of Two Chaotic QNNs

By designing an appropriate PI controller, synchronization of two chaotic QNNs with TVDs is realized in this section. The details are as follows. PI controller is designed as

$$U(t) = -K_1 e(t) - K_2 \int_{t-\tau(t)}^{t} e(s)ds, \tag{3.1}$$

where $K_1, K_2 \in \mathbb{R}^{n\times n}$ are diagonal matrices. When $e(t) \to 0$, drive-response systems achieve information synchronization, which can be ensured by Theorem 3.1.

Then, the error system (2.5) has the following form:

$$\dot{e}(t) = -\hat{D}e(t) + B\hat{f}\big(e(t)\big) + C\hat{f}\big(e\big(t-\tau(t)\big)\big) - K_2 \int_{t-\tau(t)}^{t} e(s)ds, \tag{3.2}$$

where $\hat{D} = D + K_1$.

In what follows, the stability problem of error system (2.5) will be discussed.

Theorem 3.1. *Suppose Assumption* 2.1 *holds. If there exist two real positive diagonal matrices* $\Gamma_1, \Gamma_2 \in \mathbb{R}^{n\times n}$*, two matrices* $Y, S \in \mathbb{Q}^{n\times n}$ *and four positive matrices* $W, X, Z, R \in \mathbb{Q}^{n\times n}$ *such that the following two quaternion valued linear matrix inequalities (QVLMIs) hold:*

$$\eta = \begin{pmatrix} X & Y \\ \star & Z \end{pmatrix} > 0, \tag{3.3}$$

and

$$\Xi = \begin{pmatrix} \Xi_{11} & \Xi_{12} & Y^H & SB & SC & -SK_2 \\ \star & \Xi_{22} & 0 & SB & SC & -SK_2 \\ \star & \star & \Xi_{33} & 0 & 0 & 0 \\ \star & \star & \star & -\Gamma_1 & 0 & 0 \\ \star & \star & \star & \star & -\Gamma_2 & 0 \\ \star & \star & \star & \star & \star & -\dfrac{R}{\tau} \end{pmatrix} < 0, \tag{3.4}$$

where

$$\begin{aligned} &\Xi_{11} = \tau R - S\hat{D} - \hat{D}^H S^H + L^2\Gamma_1, \quad \Xi_{12} = W - S - \hat{D}^H S^H, \\ &\Xi_{22} = \tau Z - S - S^H, \qquad\qquad\qquad \Xi_{33} = \tau X - Y - Y^H + L^2\Gamma_2, \end{aligned}$$

then error system (2.5) *with the controller* (3.1) *is globally asymptotically stable, that is, drive-response systems are synchronized.*

Proof. Construct a Lyapunov-Krasovskii functional as

$$V(t) = V_1(t) + V_2(t) + V_3(t) + V_4(t), \tag{3.5}$$

where

$$V_1(t) = e^H(t)We(t), \tag{3.6}$$

$$V_2(t) = \int_0^t \int_{u-\tau(u)}^u \begin{pmatrix} e\big(u-\tau(u)\big) \\ \dot{e}(s) \end{pmatrix}^H \begin{pmatrix} X & Y \\ \star & Z \end{pmatrix} \times \begin{pmatrix} e\big(u-\tau(u)\big) \\ \dot{e}(s) \end{pmatrix} dsdu, \tag{3.7}$$

$$V_3(t) = \int_{-\tau}^0 \int_{t+u}^t \dot{e}^H(s)Z\dot{e}(s)dsdu, \tag{3.8}$$

$$V_4(t) = \int_{-\tau}^0 \int_{t+u}^t e^H(s)Re(s)dsdu. \tag{3.9}$$

The derivatives of $V_i,\ i = 1, 2, 3, 4$, are as follows:

$$\dot{V}_1(t) = \dot{e}^H(t)We(t) + e^H(t)W\dot{e}(t), \tag{3.10}$$

$$\begin{aligned} \dot{V}_2(t) &= \tau(t)e^H\big(t-\tau(t)\big)Xe\big(t-\tau(t)\big) + e^H\big(t-\tau(t)\big)Ye(t) + e^H(t)Y^He\big(t-\tau(t)\big) \\ &\quad - e^H\big(t-\tau(t)\big)\big(Y+Y^H\big)e\big(t-\tau(t)\big) + \int_{t-\tau(t)}^t \dot{e}^H(s)Z\dot{e}(s)ds \\ &\le e^H\big(t-\tau(t)\big)\big(\tau X - Y - Y^H\big)e\big(t-\tau(t)\big) + e^H\big(t-\tau(t)\big)Ye(t) \\ &\quad + e^H(t)Y^He\big(t-\tau(t)\big) + \int_{t-\tau}^t \dot{e}^H(s)Z\dot{e}(s)ds, \end{aligned} \tag{3.11}$$

$$\begin{aligned}\dot{V}_3(t) &= \tau\dot{e}^H(t)Z\dot{e}(t) - \int_{-\tau}^{0}\dot{e}^H(t+u)Z\dot{e}(t+u)du \\ &= \tau\dot{e}^H(t)Z\dot{e}(t) - \int_{t-\tau}^{t}\dot{e}^H(s)Z\dot{e}(s)ds, \qquad (3.12)\\ \dot{V}_4(t) &= \tau e^H(t)Re(t) - \int_{t-\tau}^{t}e^H(s)Re(s)ds \\ &= \tau e^H(t)Re(t) - \frac{1}{\tau}\left[\int_{t-\tau}^{t}e(s)ds\right]^H R\left[\int_{t-\tau}^{t}e(s)ds\right]. \qquad (3.13)\end{aligned}$$

Based on (3.10)-(3.13), one can obtain

$$\begin{aligned}\dot{V}(t) \le{}& \dot{e}^H(t)We(t) + e^H(t)W\dot{e}(t) + e^H\big(t-\tau(t)\big)\big(\tau X - Y - Y^H\big)e\big(t-\tau(t)\big)\\ &+ e^H\big(t-\tau(t)\big)Ye(t) + e^H(t)Y^He\big(t-\tau(t)\big) + \tau\dot{e}^H(t)Z\dot{e}(t) + \tau e^H(t)Re(t)\\ &- \frac{1}{\tau}\left[\int_{t-\tau}^{t}e(s)ds\right]^H R\left[\int_{t-\tau}^{t}e(s)ds\right]. \qquad (3.14)\end{aligned}$$

By using the free weighting matrix method, it follows from (3.2) that

$$0 = \big[S^He(t) + S^H\dot{e}(t)\big]^HM + M^H\big[S^He(t) + S^H\dot{e}(t)\big],$$

where

$$M = -\dot{e}(t) - \hat{D}e(t) + B\hat{f}\big(e(t)\big) + C\hat{f}\big(e\big(t-\tau(t)\big)\big) - K_2\int_{t-\tau}^{t}e(s)ds.$$

From the above discussions, one obtains

$$\begin{aligned}0 = {}& -e^H(t)\big(S\hat{D} + \hat{D}^HS^H\big)e(t) - e^H(t)\big(S + \hat{D}^HS^H\big)\dot{e}(t)\\ &- \dot{e}^H(t)\big(S^H + S\hat{D}\big)e(t) + e^H(t)SB\hat{f}\big(e(t)\big) + \hat{f}^H\big(e(t)\big)B^HS^He(t)\\ &+ e^H(t)SC\hat{f}\big(e\big(t-\tau(t)\big)\big) + \hat{f}\big(e\big(t-\tau(t)\big)\big)C^HS^He(t)\\ &- e^H(t)SK_2\left(\int_{t-\tau}^{t}e(s)ds\right) - \left(\int_{t-\tau}^{t}e(s)ds\right)^HK_2^HS^He(t)\\ &- \dot{e}^H(t)\big(S + S^H\big)\dot{e}(t) + \dot{e}^H(t)SB\hat{f}\big(e(t)\big)\\ &+ \hat{f}^H\big(e(t)\big)B^HS^H\dot{e}(t) + \dot{e}^H(t)SC\hat{f}\big(e\big(t-\tau(t)\big)\big)\\ &+ \hat{f}^H\big(e\big(t-\tau(t)\big)\big)C^HS^H\dot{e}(t) - \dot{e}^H(t)SK_2\left(\int_{t-\tau}^{t}e(s)ds\right)\\ &- \left(\int_{t-\tau}^{t}e(s)ds\right)^HK_2^HS^H\dot{e}(t). \qquad (3.15)\end{aligned}$$

Additionally, based on the assumption 2.1, one knows that the following inequalities hold:

$$\begin{aligned}&e^H(t)L^2\Gamma_1e(t) - \hat{f}^H\big(e(t)\big)\Gamma_1\hat{f}\big(e(t)\big) \ge 0,\\ &e^H\big(t-\tau(t)\big)L^2\Gamma_2e\big(t-\tau(t)\big) - \hat{f}^H\big(e\big(t-\tau(t)\big)\big)\Gamma_2\hat{f}\big(e\big(t-\tau(t)\big)\big) \ge 0.\end{aligned} \qquad (3.16)$$

From (3.14)-(3.16), the following inequality holds:

$$\dot{V}(t) \le \mu^H(t)\Pi\mu(t), \qquad (3.17)$$

where

$$\mu(t) = \left[e^H(t), \dot{e}^H(t), e^H\big(t-\tau(t)\big), \hat{f}^H\big(e(t)\big), \hat{f}^H\big(e\big(t-\tau(t)\big)\big), \int_{t-\tau}^{t} e^H(s)ds\right]^H,$$

and

$$\Pi = \begin{pmatrix} \Pi_{11} & \Pi_{12} & Y^H & SB & SC & -SK_2 \\ \star & \Pi_{22} & 0 & SB & SC & -SK_2 \\ \star & \star & \Pi_{33} & 0 & 0 & 0 \\ \star & \star & \star & -\Gamma_1 & 0 & 0 \\ \star & \star & \star & \star & -\Gamma_2 & 0 \\ \star & \star & \star & \star & \star & -\dfrac{R}{\tau} \end{pmatrix}, \tag{3.18}$$

where

$$\begin{aligned} &\Pi_{11} = \tau R - S\hat{D} - \hat{D}^H S^H + L^2\Gamma_1, \quad \Pi_{12} = W - S - \hat{D}^H S^H, \\ &\Pi_{22} = \tau Z - S - S^H, \qquad\qquad\qquad\quad \Pi_{33} = \tau X - Y - Y^H + L^2\Gamma_2. \end{aligned}$$

It is easy to verify $\Pi < 0$ and $\Omega < 0$. Thus, it follows from (3.4) and (3.17) that $\dot{V}(t) < 0$. Based on the standard Lyapunov theorem, we know that the error system (3.2) is globally asymptotically stable. The proof is complete. □

Remark 3.1. QVLMIs (3.3) and (3.4) cannot be directly handled via the Matlab LMI toolbox. As mentioned in [8], QVLMIs can be transformed into complex-valued ones. Complex-valued LMIs can be checked directly by the YALMIP toolbox in Matlab. Thus, the proof is omitted.

Remark 3.2. For complex-valued or QNNs, scholars and researchers usually decompose them into real-valued systems. The decomposed real-valued systems are equivalent to original systems. But it will increase the dimension of the studied systems. Then, QNNs do not be decomposed into four real-valued systems in this paper. As mentioned in [8], based on QVLMI method and free weighting matrix technique, the stability of quaternion-valued error systems (2.5) is verified, that is, the synchronization of drive chaotic systems (2.2) and response chaotic systems (2.4) is realized.

4. Simulation and Application to Image Encryption and Decryption

In this section, two numerical cases are presented to verify the effectiveness of the above results.

4.1. Case 1

Example 4.1. Consider two neurons neural network as drive-response systems, which are as follows:

$$\dot{x}(t) = -Dx(t) + Bf\big(x(t)\big) + Cf\big(x\big(t-\tau(t)\big)\big) + I, \tag{4.1}$$

$$\dot{y}(t) = -Dy(t) + Bf\big(y(t)\big) + Cf\big(y\big(t-\tau(t)\big)\big) + I + U, \tag{4.2}$$

where

$$f\big(x(t)\big) = \tanh\big(x^R(t)\big) + \boldsymbol{i}\tanh\big(x^I(t)\big) + \boldsymbol{j}\tanh\big(x^J(t)\big) + \boldsymbol{k}\tanh\big(x^K(t)\big), \quad \tau(t) = 1.2, \quad I = 0,$$

$$x = \begin{pmatrix} x_1^R + x_1^I\boldsymbol{i} + x_1^J\boldsymbol{j} + x_1^K\boldsymbol{k} \\ x_2^R + x_2^I\boldsymbol{i} + x_2^J\boldsymbol{j} + x_2^K\boldsymbol{k} \end{pmatrix},$$

$$y = \begin{pmatrix} y_1^R + y_1^I\boldsymbol{i} + y_1^J\boldsymbol{j} + y_1^K\boldsymbol{k} \\ y_2^R + y_2^I\boldsymbol{i} + y_2^J\boldsymbol{j} + y_2^K\boldsymbol{k} \end{pmatrix},$$

$$D = \begin{pmatrix} 1 & \\ & 1 \end{pmatrix},$$

$$B = \begin{pmatrix} -0.4 - 0.3\boldsymbol{i} - 0.3\boldsymbol{j} + 0.4\boldsymbol{k} & -0.6 - 0.1\boldsymbol{i} - 0.1\boldsymbol{j} + 0.6\boldsymbol{k} \\ -1.5 - 0.6\boldsymbol{i} + 0.2\boldsymbol{j} + 0.4\boldsymbol{k} & -0.1 + 0.3\boldsymbol{i} + 0.3\boldsymbol{j} + 0.6\boldsymbol{k} \end{pmatrix},$$

$$C = \begin{pmatrix} 2 + 0.5\boldsymbol{i} - 0.3\boldsymbol{j} + 0.3\boldsymbol{k} & 1.6 - 0.4\boldsymbol{i} - 0.2\boldsymbol{j} - 0.3\boldsymbol{k} \\ 0.3 + 0.1\boldsymbol{i} + 0.2\boldsymbol{j} + 0.2\boldsymbol{k} & -2.5 - 0.3\boldsymbol{i} + 0.4\boldsymbol{j} - 0.5\boldsymbol{k} \end{pmatrix}.$$

The initial conditions are

$$x(s) = \begin{pmatrix} 1 + 0.5\boldsymbol{i} + \boldsymbol{j} + \boldsymbol{k} \\ -1 + \boldsymbol{i} - 1.2\boldsymbol{j} - \boldsymbol{k} \end{pmatrix}, \quad y(s) = \begin{pmatrix} -1 - 0.5\boldsymbol{i} - \boldsymbol{j} - \boldsymbol{k} \\ 1 - \boldsymbol{i} + 1.2\boldsymbol{j} + \boldsymbol{k} \end{pmatrix}.$$

Let $K_1 = \mathrm{diag}(6, 8)$, $K_2 = \mathrm{diag}(0.1, 0.01)$, and the PI controller $U = 0$. State trajectories of drive system (4.1) are shown in Figs. 4.1 and 4.2. Phase plot of drive system (4.1) is given in Fig. 4.3. And state trajectories of response system (4.2) are shown in Figs. 4.4 and 4.5. Phase plot of response system (4.2) is given in Fig. 4.3. By Figs. 4.3 and 4.6, one can see that drive-response systems are chaotic. Time response of error system is shown in Fig. 4.7.

When $U \neq 0$, according to Theorem 3.1, systems (4.1) and (4.2) can achieve data synchronization under controller (3.1). From Figs. 4.8 and 4.9, one can observe that the drive system (4.1) and the response system (4.2) is synchronized via controller (3.1).

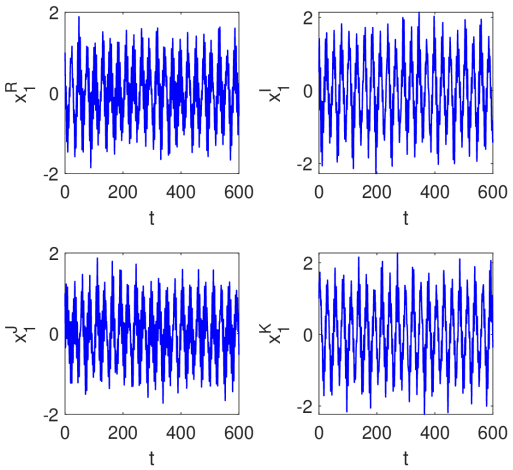


Fig. 4.1. State trajectory of $x_1(t)$.

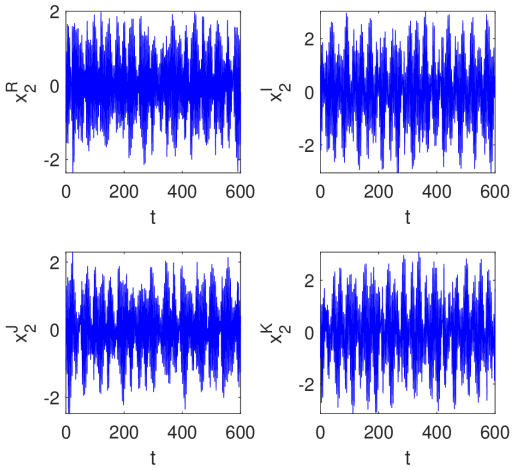


Fig. 4.2. State trajectory of $x_2(t)$.

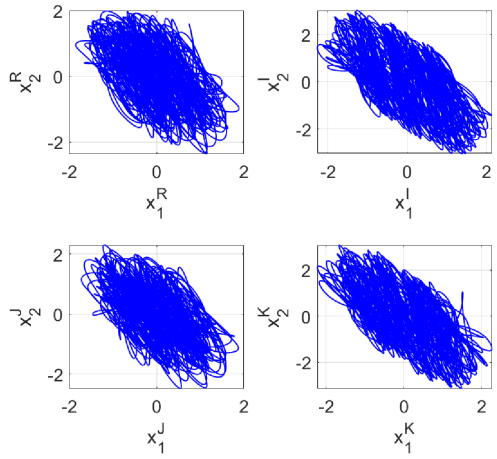


Fig. 4.3. Phase plot of drive system (4.1).

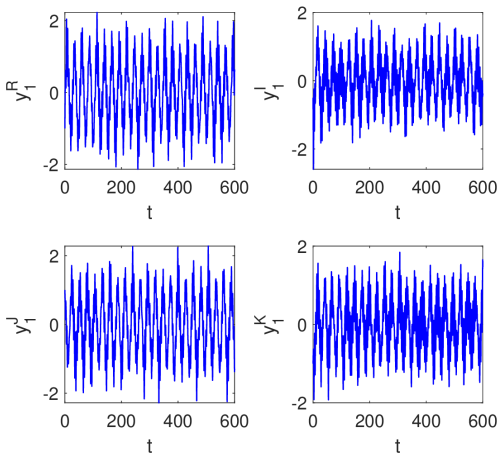


Fig. 4.4. State trajectory of $y_1(t)$ without controller.

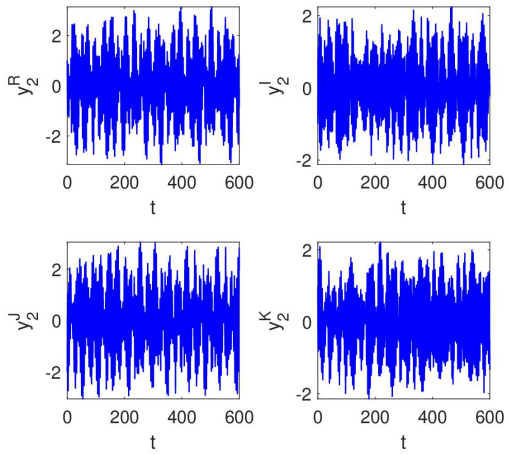


Fig. 4.5. State trajectory of $y_2(t)$ without controller.

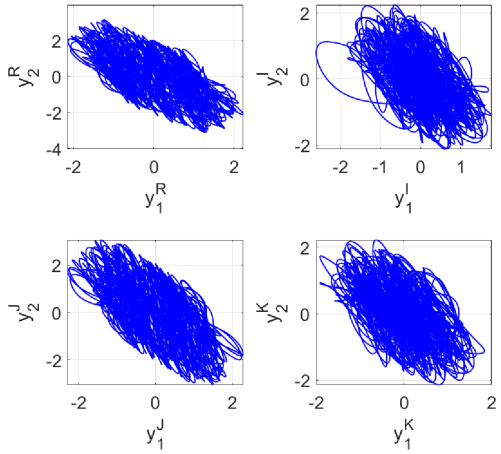
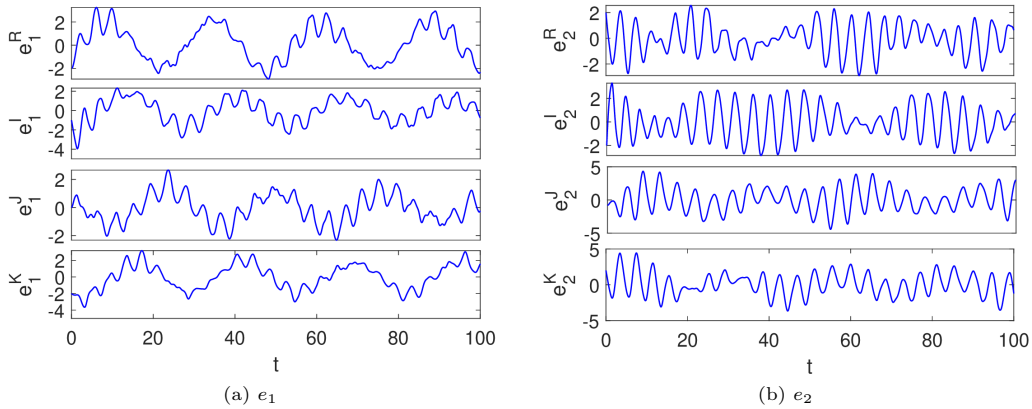


Fig. 4.6. Phase plot of response system (4.1) without controller.



(a) e_1 (b) e_2

Fig. 4.7. Time response of error system without controller.

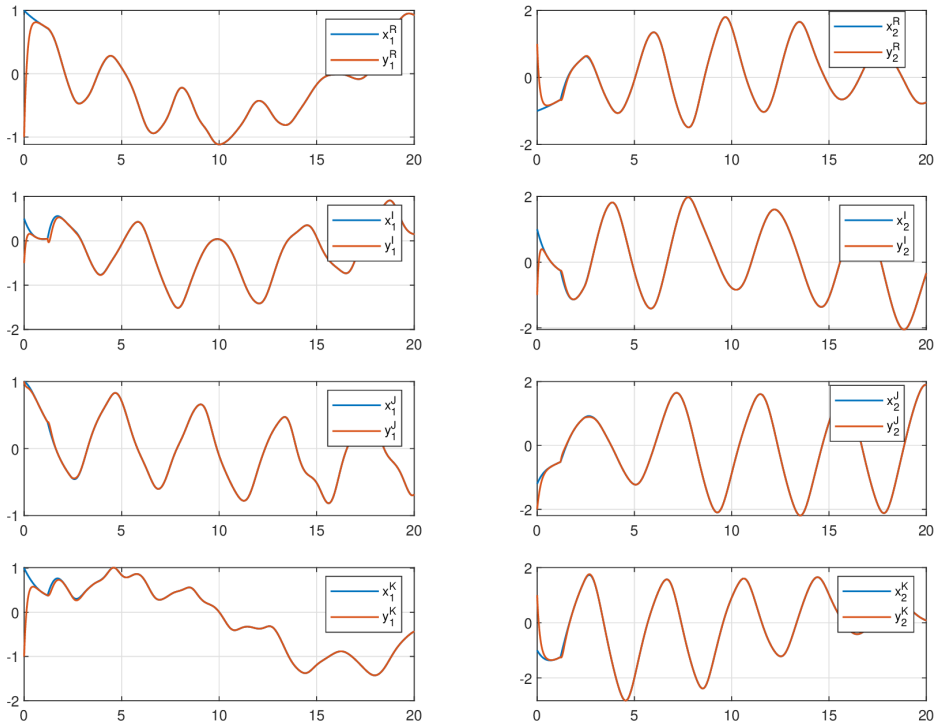
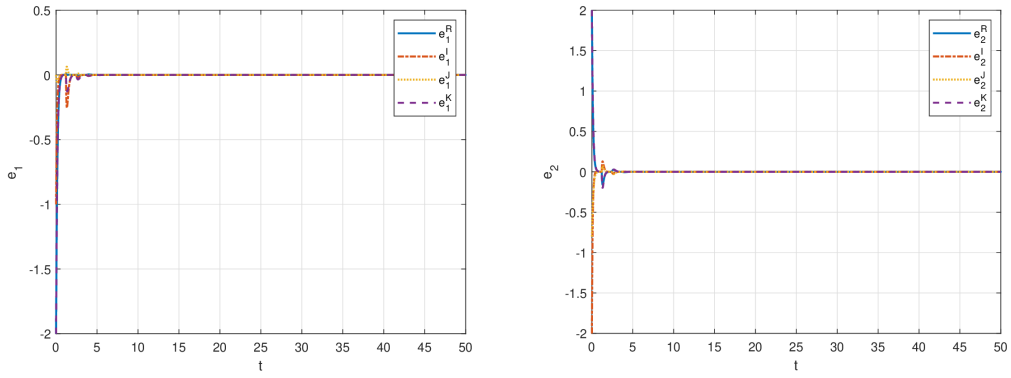


Fig. 4.8. Time responses of drive-response systems with PI controller.



(a) e_1 (b) e_2

Fig. 4.9. Time responses of error system with PI controller.

4.2. Case 2

Example 4.2. Using the results obtained in Example 4.1, application of chaotic QNNs with TVDs in image encryption and decryption is carried out. Based on Example 4.1, one sequence of chaotic signal is obtained using state trajectory of single quaternion-valued neuron, such as $\{z_i(t_k) : k = 1, 2, \ldots\}$, where $z_i = x_i^R + \boldsymbol{i}x_i^I + \boldsymbol{j}x_i^J + \boldsymbol{k}x_i^K$. $z_i(t_k) : k = 1, 2, \ldots$ will serve as encryption signals. Details of encryption algorithm are as follows.

The process of encryption is concluded and the ciphered image is obtained. The decryption algorithm is the reverse process of encryption algorithm. Here, it is omitted.

Algorithm 4.1: Encryption Algorithm.

Step 1: Read original image $\mathfrak{M}$. Separating color image $\mathfrak{M}$ with red, green and blue components, respectively. Three pixel series are obtained as $\mathfrak{M}_R(i,j)$, $\mathfrak{M}_G(i,j)$, $\mathfrak{M}_B(i,j)$, $i \in \{1,2,\ldots,N\}$, $j \in \{1,2,\ldots,N\}$.

Step 2: Through driving system (4.1), group of time series chaotic signals $\{x_i\ x_i^I\ x_i^J\ x_i^K\}$ are obtained by using `Matlab dde23` algorithm and the initial conditions $z_j(0)$.

Step 3: Time series chaotic signals x_1, x_1^I, x_1^J and x_1^K are represented by matrix form X_1, X_1^I, X_1^J and X_1^K

$$\begin{aligned} X_1(i,j) &= x_1(k), \quad X_1^I(i,j) = x_1^I(k), \\ X_1^J(i,j) &= x_1^J(k), \quad X_1^K(i,j) = x_1^K(k), \end{aligned}$$

where $i \in \{1,2,\ldots,N\}$, $j \in \{1,2,\ldots,N\}$, $k \in \{1,2,\ldots,N \times N\}$.

Step 4: Further from the above matrices, three new matrices X_R, X_G and X_B are generated as

$$\begin{aligned} X_R(i,j) &= \operatorname{mod}\big(\operatorname{round}\big(X_1^I(i,j),k\big) \times\ 10^9, 256\big), \\ X_G(i,j) &= \operatorname{mod}\big(\operatorname{round}\big(X_1^J(i,j),k\big) \times\ 10^{9.5}, 256\big), \\ X_B(i,j) &= \operatorname{mod}\big(\operatorname{round}\big(X_1^K(i,j),k\big) \times\ 10^9, 256\big), \end{aligned}$$

where $\operatorname{mod}(X,Y)$ represents the remainder after division, $\operatorname{round}(X,k)$ denotes the operation of rounding to k digits to the right of the decimal point.

Step 5: Set $n = 1$, permute images $\mathfrak{M}_R, \mathfrak{M}_G, \mathfrak{M}_B$. According to Arnold cat map permutation algorithm, $\mathfrak{M}_R, \mathfrak{M}_G, \mathfrak{M}_B$ are transferred to permuted images R', G', B'. Three new pixel series are obtained as $R'(i,j), G'(i,j), B'(i,j)$, $i \in \{1,2,\ldots,N\}$, $j \in \{1,2,\ldots,N\}$.

Step 6: The permuted images R', G' and B' are encrypted by X_R, X_G and X_B to obtain $\mathfrak{M}_R, \mathfrak{M}_G$ and $\mathfrak{M}_B$ as

$$\begin{aligned} \mathfrak{M}_R(i,j) &= R'(i,j) \oplus x_R(i,j), \\ \mathfrak{M}_G(i,j) &= G'(i,j) \oplus x_G(i,j), \\ \mathfrak{M}_B(i,j) &= B'(i,j) \oplus x_B(i,j), \end{aligned}$$

where $\oplus$ denotes the bitwise XOR operator.

Step 7: If $n < R$, set $n = n + 1$ return to Step 5. Otherwise, encryption process completes.

Step 8: Reorganizing $\mathfrak{M}_R, \mathfrak{M}_G$ and $\mathfrak{M}_B$, we can obtain the encrypted color image.

Statistical analysis

The performance of the above encryption algorithm is investigated by Statistical analyses, including histogram analysis, information entropy and correlation coefficient.

(a) Histogram analysis: As shown in Fig. 4.10, the histograms of the encrypted images are quite uniform and have good statistical properties similar to white noise. In this way, information about the pixel order in the original image cannot be obtained from the encrypted image.

(b) Information entropy analysis: Information entropy is one of the criteria for measuring the strength of cryptographic systems. Information entropies of the three color components (R,G,B) of Fig. 4.10(b) are compared with the other algorithms in Table 4.1. Compared to the other algorithms, our scheme has the highest entropy, which is very close to the ideal value of 8. Therefore, encrypted image is close to random sources, and the algorithm proposed is secure for entropy attacks.

(c) Correlation coefficient analysis: In Fig. 4.11, the correlations between horizontally adjacent pixels in three color components (R,G,B) of plain-images and encrypted images are compared. From Fig. 4.11, it can be observed that although the two adjacent pixels in the plain-images are highly correlated, the correlations between these two adjacent pixels in the encrypted image can be ignored. From Table 4.2, it can be seen that the proposed image encryption algorithm has better performance compared with other algorithms. Thus, the correlation of encrypted Lena images is very low, indicating that our encryption algorithm has good encryption performance.

Remark 4.1. Regarding the practicality of the image encryption and decryption algorithm presented in this article: From an implementation perspective, it is designed to be straightforward and can be easily adapted to different software and hardware platforms. Detailed step-by-step instructions and examples are provided to facilitate its implementation in real-world applications. In terms of computation, the algorithm is optimized to ensure efficient

Table 4.1: Information entropies of the encrypted-images with different encryption algorithms.

Scheme	H (Red)	H (Green)	H (Blue)
[49]	7.6782	7.8402	5344
[26]	7.9896	7.9893	7.9896
[28]	7.9972	7.9973	7.9972
[41]	7.9971	7.9972	7.9965
Proposed algorithm	7.9977	7.9975	7.9975

Table 4.2: Correlation of the encrypted-image.

Scheme	H (Red)	H (Green)	H (Blue)
[49]	0.0158	0.0222	0.0484
[41]	0.0048	0.0051	0.0043
Proposed algorithm	-0.0036	-0.0030	0.0017

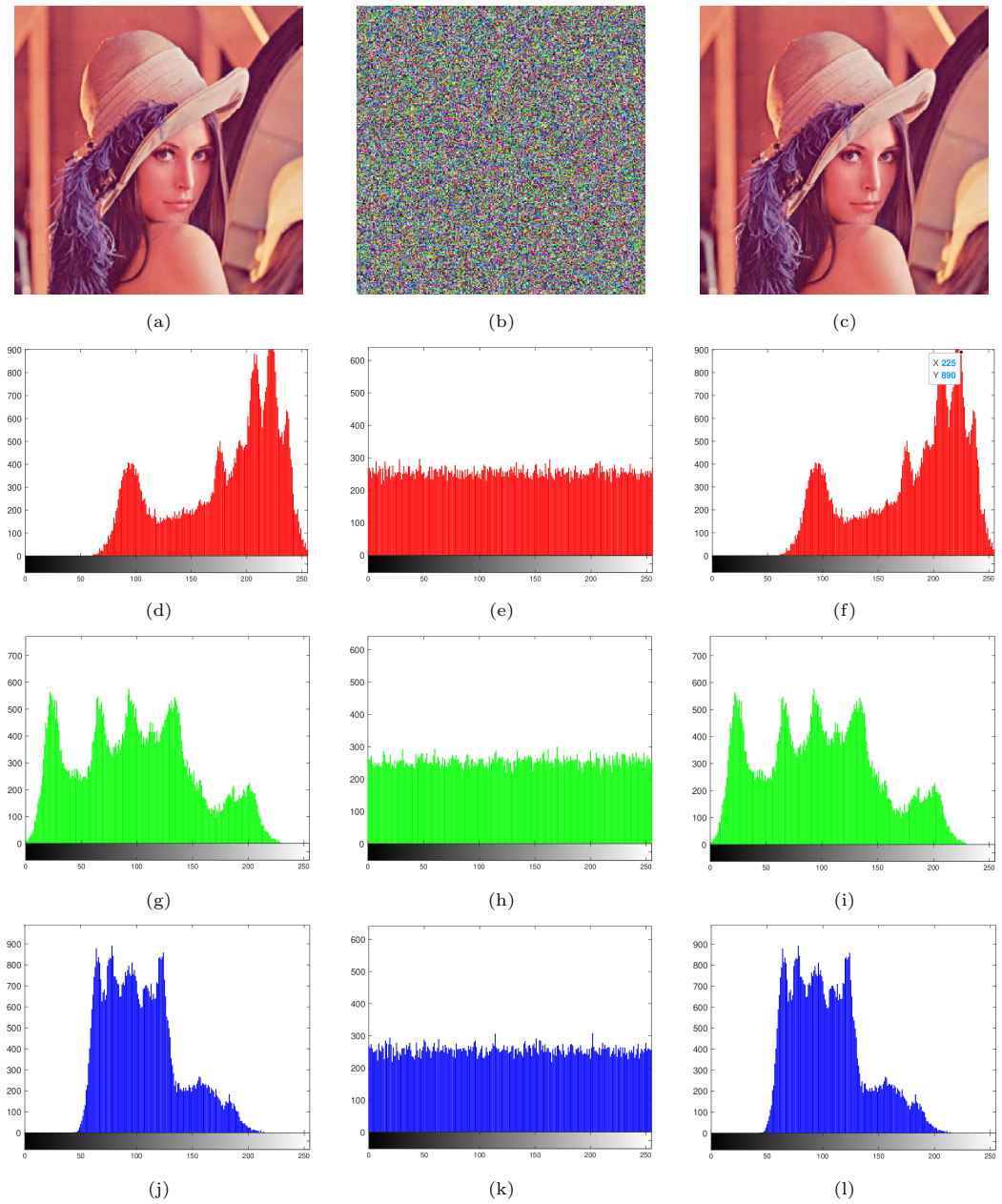


Fig. 4.10. Results of image encryption and decryption. (a) Original image. (b) Shuffled image. (c) Decrypted image: succeeded. Histogram of original image: (d) red color, (g) green color, (j) blue color. Histogram of encrypted image: (e) red color, (h) green color, (k) blue color. Histogram of decrypted image: (f) red color, (i) green color, (l) blue color.

processing. It makes use of efficient computational techniques to minimize the time and resources required. The computational complexity is well within acceptable limits, and it can be effectively executed on standard computing devices. Concerning the level of computational difficulty, while it does involve some computations, they are not overly complex. The key operations are well-structured and can be efficiently carried out.

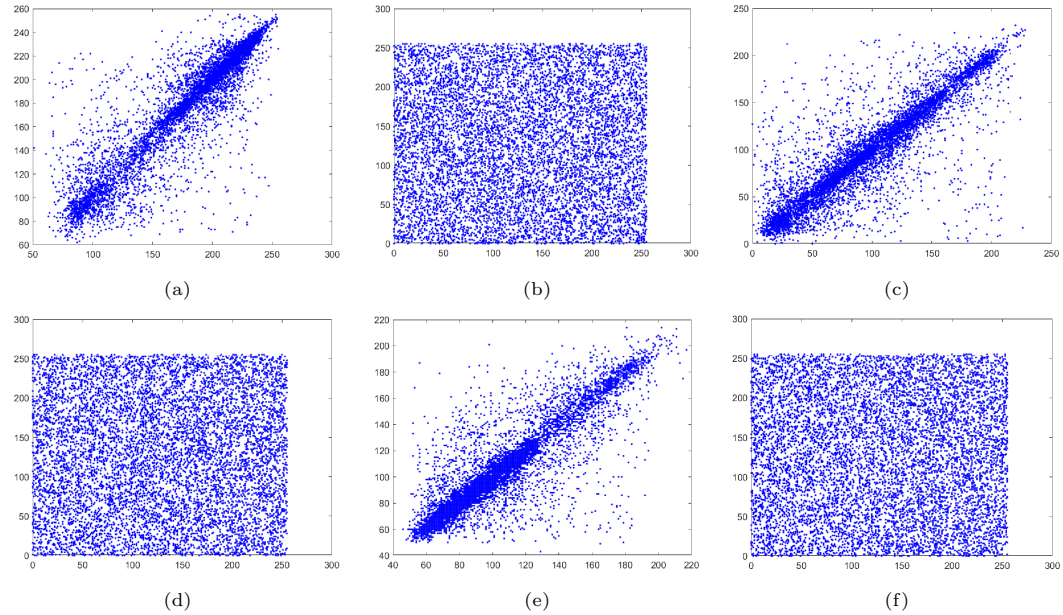


Fig. 4.11. Correlation plot of two adjacent plain-image pixels in horizontal direction: (a) red color, (c) green color, (e) blue color. Correlation plot of two adjacent pixels of the cipher-image obtained by the proposed scheme in horizontal direction: (b) red color, (d) green color, (f) blue color.

5. Conclusions

This paper researched the synchronization problem of chaotic QNNs with TVDs and its application in cryptosystem. On one hand, by designing a suitable PI controller, synchronization of the proposed system was realized. By using non-decomposition method and LF, sufficient conditions was derived to ensure synchronization of drive-response systems. Here, QNNs do not be decomposed into four real-valued systems. On the other hand, application of chaotic QNNs with TVDs in image encryption and decryption was carried out. Based on one sequence of chaotic signal from state trajectory of single quaternionic neuron and a new encryption algorithm, the application of chaotic system proposed, that is, image encryption, was researched. The image decryption process was the reverse process of image encryption. Finally, numerical simulations were performed to demonstrate the validity of the obtained results.

Acknowledgments. This research was funded by the Scientific Research Program Funded by the Shaanxi Provincial Education Department (Grant No. 23JK0550), by the Natural Science Basic Research Program of Shaanxi Province (Grant Nos. 2024JC-YBQN-0726, 2024JC-YBMS-352 and 2023-JC-QN-0001), by the Qin Chuangyuan "Scientists + Engineers" Team Construction in Shaanxi Province (Grant No. 2022KXJ-38), by the Xi'an Science and Technology Program (Grant No. 2022JH-RGZN-0041), and by the Xi'an Science and Technology Program (Grant No. 23ZDCYJSGG0025-2022).

References

[1] H. Abbasi, M. Yaghoobi, M. Teshnehlab, and A. Sharifi, Cascade chaotic neural network (CCNN): A new model, *Neural. Comput. Appl.*, **34** (2022), 8897–8917.

[2] N. Abdoun, S.E. Assad, O. Déforges, R. Assaf, and M. Khalil, Design and security analysis of two robust keyed hash functions based on chaotic neural networks, *J. Ambient Intell. Humaniz. Comput.*, **11**:5 (2020), 2137–2161.

[3] K. Aihara, Chaos engineering and its application to parallel distributed processing with chaotic neural networks, *Proc. IEEE*, **90**:5 (2002), 919–930.

[4] N. Bigdeli, Y. Farid, and K. Afshar, A novel image encryption/decryption scheme based on chaotic neural networks, *Eng. Appl. Artif. Intell.*, **25**:4 (2012), 753–765.

[5] N. Bigdeli, Y. Farid, and K. Afshar, A robust hybrid method for image encryption based on Hopfield neural network, *Comput. Electr. Eng.*, **38**:2 (2012), 356–369.

[6] A. Cariow and G. Cariowa, Fast algorithms for quaternion-valued convolutional neural networks, *IEEE Trans. Neural Netw. Learn. Syst.*, **32**:1 (2021), 457–462.

[7] X. Chen, Z. Li, Q. Song, J. Hu, and Y. Tan, Robust stability analysis of quaternion-valued neural networks with time delays and parameter uncertainties, *Neural Netw.*, **91** (2017), 55–65.

[8] X. Chen and Q. Song, State estimation for quaternion-valued neural networks with multiple time delays, *IEEE Trans. Syst. Man Cybern. Syst.*, **49**:11 (2019), 2278–2287.

[9] X. Chen, Q. Song, Z. Li, Z. Zhao, and Y. Liu, Stability analysis of continuous-time and discrete-time quaternion-valued neural networks with linear threshold neurons, *IEEE Trans. Neural Netw. Learn. Syst.*, **29**:7 (2018), 2769–2781.

[10] Y. Cheng and Y. Shi, The exponential synchronization and asymptotic synchronization of quaternion-valued memristor-based cohen-grossberg neural networks with time-varying delays, *Neural Process. Lett.*, **55** (2023), 6637–6656.

[11] J. Chou, Quaternion kinematic and dynamic differential equations, *IEEE Trans. Robot. Autom.*, **8**:1 (1992), 53–64.

[12] L. Dai and Y. Hou, Mean-square exponential input-to-state stability of stochastic quaternion-valued neural networks with time-varying delays, *Adv. Differ. Equ.*, **2021** (2021), 362.

[13] P. Guo, Q. Shi, Z. Jian, J. Zhang, Q. Ding, and W. Yan, An intelligent controller of homo-structured chaotic systems under noisy conditions and applications in image encryption, *Chaos Solitons Fractals*, **180** (2024), 114524.

[14] W. He, T. Luo, Y. Tang, W. Du, Y. Tian, and F. Qian, Secure communication based on quantized synchronization of chaotic neural networks under an event-triggered strategy, *IEEE Trans. Neural Netw. Learn. Syst.*, **31**:9 (2020), 3334–3345.

[15] C. Huang, X. Nie, X. Zhao, Q. Song, Z. Tu, M. Xiao, and J. Cao, Novel bifurcation results for a delayed fractional-order quaternion-valued neural network, *Neural Netw.*, **117** (2019), 67–93.

[16] Y. Huang, H. Huang, Y. Huang, Y. Wang, F. Yu, and B. Yu, Drive-response asymptotic shape synchronization for a class of two-dimensional chaotic systems and its application in image encryption, *Phys. D*, **463** (2024), 134162.

[17] T. Isokawa, N. Matsui, and H. Nishimura, Quaternionic neural networks: Fundamental properties and applications, in: *Proceedings of the Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*, IGI Global, 411–439, 2009.

[18] M. Kalpana, K. Ratnavelu, P. Balasubramaniam, and M.Z.M. Kamali, Synchronization of chaotic-type delayed neural networks and its application, *Nonlinear Dyn.*, **93** (2018), 543–555.

[19] L. Kocarev, G. Jakimoski, T. Stojanovski, and U. Parlitz, From chaotic maps to encryption schemes, in: *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 514–517, 1998.

[20] C. Li, J. Cao, and A. Kashkynbayev, Global finite-time stability of delayed quaternion-valued neural networks based on a class of extended Lyapunov-Razumikhin methods, *Cogn. Neurodynamics*, **17** (2022), 729–739.

[21] H. 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, *Appl. Math. Comput.*, **373** (2020), 125020.

[22] N. Li and W. Zheng, Passivity analysis for quaternion-valued memristor-based neural networks with time-varying delay, *IEEE Trans. Neural Netw. Learn. Syst.*, **31**:2 (2020), 639–650.
[23] R. Li and J. Cao, Exponential state estimation and passivity of fuzzy quaternion-valued memristive neural networks: Norm approach, *IEEE Trans. Syst. Man Cybern. Syst.*, **54**:8 (2024), 4798-4805.
[24] R. Li, J. Cao, and N. Li, Stabilization control of quaternion-valued fractional-order discrete-time memristive neural networks, *Neurocomputing*, **542** (2023), 126255.
[25] C. Liang, Q. Zhang, J. Ma, and K. Li, Research on neural network chaotic encryption algorithm in wireless network security communication, *EURASIP J. Wirel. Commun. Netw.*, **2019** (2019), 151.
[26] H. Liu and A. Kadir, Asymmetric color image encryption scheme using 2D discrete-time map, *Signal Process.*, **113** (2015), 104–112.
[27] G. Maddodi, A. Awad, D. Awad, M. Awad, and B. Lee, A new image encryption algorithm based on heterogeneous chaotic neural network generator and dna encoding, *Multimed. Tools. Appl.*, **77** (2018), 24701–24725.
[28] A.Y. Niyat, H.M. Mohammad, and M.N. Torshiz, Color image encryption based on hybrid hyper-chaotic system and cellular automata, *Opt. Lasers Eng.*, **90** (2017), 225–237.
[29] K. Nosrati, C. Volos, and A. Azemi, Cubature Kalman filter-based chaotic synchronization and image encryption, *Signal Process. Image Commun.*, **58** (2017), 35–48.
[30] S.M.A. Pahnehkolaei, A. Alfi, and J.T. Machado, Delay independent robust stability analysis of delayed fractional quaternion-valued leaky integrator echo state neural networks with QUAD condition, *Appl. Math. Comput.*, **359** (2019), 278–293.
[31] S.M.A. Pahnehkolaei, A. Alfi, and J.T. Machado, Delay-dependent stability analysis of the QUAD vector field fractional order quaternion-valued memristive uncertain neutral type leaky integrator echo state neural networks, *Neural Netw.*, **117** (2019), 307–327.
[32] C. Pan, Q. Hong, and X. Wang, A novel memristive chaotic neuron circuit and its application in chaotic neural networks for associative memory, *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, **40**:3 (2021), 521–532.
[33] C. Popa, Learning algorithms for quaternion-valued neural networks, *Neural Process. Lett.*, **47** (2018), 949–973.
[34] G. Rajchakit, P. Agarwal, and S. Ramalingam, *Stability Analysis of Neural Networks*, Springer, 2021.
[35] G. Rajchakit, P. Chanthorn, P. Kaewmesri, R. Sriraman, and C.P. Lim, Global Mittag-Leffler stability and stabilization analysis of fractional-order quaternion-valued memristive neural networks, *Mathematics*, **8**:3 (2020), 422.
[36] G. Rajchakit and R. Sriraman, Robust passivity and stability analysis of uncertain complex-valued impulsive neural networks with time-varying delays, *Neural Process. Lett.*, **53** (2021), 581–606.
[37] G. Rajchakit, R. Sriraman, N. Boonsatit, P. Hammachukiattikul, C.P. Lim, and P. Agarwal, Exponential stability in the Lagrange sense for Clifford-valued recurrent neural networks with time delays, *Adv. Contin. Discrete Models*, **2021** (2021), 256.
[38] G. Rajchakit, R. Sriraman, C.P. Lim, and B. Unyong, Existence, uniqueness and global stability of Clifford-valued neutral-type neural networks with time delays, *Math. Comput. Simulation*, **201** (2022), 508–527.
[39] S. Ramalingam and O. Kwon, Non-separation method-based global stability criteria for Takagi-Sugeno fuzzy quaternion-valued BAM delayed neural networks using quaternion-valued auxiliary function-based integral inequality, *Neural Process. Lett.*, **56** (2024), 101.
[40] L.S. Saoud, R. Ghorbani, and F. Rahmoune, Cognitive quaternion valued neural network and some applications, *Neurocomputing*, **221** (2017), 85–93.
[41] L. Shanmugam, P. Mani, R. Rajan, and Y.H. Joo, Adaptive synchronization of reaction diffusion neural networks and its application to secure communication, *IEEE Trans. Cybern.*, **50**:3 (2020), 911–922.

[42] Q. Song, L. Yang, Y. Liu, and F.E. Alsaadi, Stability of quaternion-valued neutral-type neural networks with leakage delay and proportional delays, *Neurocomputing*, **521** (2023), 191–198.

[43] G. Tan, Z. Wang, and Z. Shi, Proportional-integral state estimator for quaternion-valued neural networks with time-varying delays, *IEEE Trans. Neural Netw. Learn. Syst.*, **34**:2 (2023), 1074–1079.

[44] X. Tan, C. Xiang, J. Cao, W. Xu, G. Wen, and L. Rutkowski, Synchronization of neural networks via periodic self-triggered impulsive control and its application in image encryption, *IEEE Trans. Cybern.*, **52**:8 (2022), 8246–8257.

[45] C. Volos, I. Kyprianidis, and I. Stouboulos, Image encryption process based on chaotic synchronization phenomena, *Signal Process.*, **93**:5 (2013), 1328–1340.

[46] Y. Wan, J. Cao, and G. Wen, Quantized synchronization of chaotic neural networks with scheduled output feedback control, *IEEE Trans. Neural Netw. Learn. Syst.*, **28**:11 (2017), 2638–2647.

[47] P. Wang, X. Li, N. Wang, Y. Li, K. Shi, and J. Lu, Almost periodic synchronization of quaternion-valued fuzzy cellular neural networks with leakage delays, *Fuzzy Sets and Systems*, **426** (2022), 46–65.

[48] W. Wei, J. Yu, L. Wang, C. Hu, and H. Jiang, Fixed/preassigned-time synchronization of quaternion-valued neural networks via pure power-law control, *Neural Netw.*, **146** (2022), 341–349.

[49] A. Wu, Y. Chen, and Z. Zeng, Quantization synchronization of chaotic neural networks with time delay under event-triggered strategy, *Cogn. Neurodynamics*, **15** (2021), 897–914.

[50] G. Zhang and Y. Shen, Exponential stabilization of memristor-based chaotic neural networks with time-varying delays via intermittent control, *IEEE Trans. Neural Netw. Learn. Syst.*, **26**:7 (2015), 1431–1441.

[51] H. Zhang, C. Wang, W. Zhang, and H. Zhang, Mittag-Leffler stability and synchronization for FO-QVFNNs including proportional delay and Caputo derivative via fractional differential inequality approach, *Comput. Appl. Math.*, **41** (2022), 344.

[52] Q. Zhang, X. Wei, and J. Xu, Asymptotic stability of transiently chaotic neural networks, *Nonlinear Dyn.*, **42** (2005), 339–346.

[53] Y. Zhao, X. Li, and P. Duan, Observer-based sliding mode control for synchronization of delayed chaotic neural networks with unknown disturbance, *Neural Netw.*, **117** (2019), 268–273.

[54] J. Zhu and J. Sun, Stability of quaternion-valued impulsive delay difference systems and its application to neural networks, *Neurocomputing*, **284** (2018), 63–69.