

Complex Rotation Quantum Dynamic Neural Networks (CRQDNN) using Complex Quantum Neuron (CQN): Applications to time series prediction



Yiqian Cui^{*}, Junyou Shi, Zili Wang

School of Reliability and Systems Engineering, Beihang University, Beijing, China

Science and Technology Key Laboratory on Reliability and Environmental Engineering, Beihang University, Beijing, China

HIGHLIGHTS

- A novel hybrid networks model Complex Rotation Quantum Dynamic Neural Networks (CRQDNN) is proposed.
- Deep quantum entanglement is realized by incorporating Complex Quantum Neurons (QRN).
- The embedded IIR filter structure enables the dynamic properties to treat with time series input.
- The application studies of chaotic time series prediction and electronic prognostics are investigated.

ARTICLE INFO

Article history:

Received 3 December 2014

Received in revised form 2 July 2015

Accepted 23 July 2015

Available online 31 July 2015

Keywords:

Quantum entanglement
Complex Quantum Neuron (CQN)
Infinite Impulse Response (IIR)
Chaotic time series prediction
Remaining Useful Life (RUL) prediction

ABSTRACT

Quantum Neural Networks (QNN) models have attracted great attention since it innovates a new neural computing manner based on quantum entanglement. However, the existing QNN models are mainly based on the real quantum operations, and the potential of quantum entanglement is not fully exploited. In this paper, we proposes a novel quantum neuron model called Complex Quantum Neuron (CQN) that realizes a deep quantum entanglement. Also, a novel hybrid networks model Complex Rotation Quantum Dynamic Neural Networks (CRQDNN) is proposed based on Complex Quantum Neuron (CQN). CRQDNN is a three layer model with both CQN and classical neurons. An infinite impulse response (IIR) filter is embedded in the Networks model to enable the memory function to process time series inputs. The Levenberg–Marquardt (LM) algorithm is used for fast parameter learning. The networks model is developed to conduct time series predictions. Two application studies are done in this paper, including the chaotic time series prediction and electronic remaining useful life (RUL) prediction.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Time series prediction refers to the study of the present and past behavior of the system for the prediction of the future (Widiputra, Pears, & Kasabov, 2011). Artificial Neural Networks (ANN) are universally employed in time series prediction since they have the properties such as self-organizing, data-driven, self-study, self-adaptive and associated memory (Zhang, Eddy Patuwo, & Y Hu, 1998).

Traditional static ANN models do not incorporate the temporal cumulative effect because a single input sample is either irrelevant to time or relative to a moment instead of a period of time.

The time-varying effect is supposed to be incorporated in neural networks models (Kim, 1997). To better deal with the time series inputs, from about 1990s, great attention has been paid to the development of Dynamic Neural Networks (DNN) due to their capabilities in modeling nonlinear dynamical systems (Lou & Cui, 2007; Meyer-Bäse, Ohl, & Scheich, 1996).

Recently, Quantum Neural Networks (QNN) models have attracted great attention worldwide because of its novel computational characteristics. Quantum entanglement is involved in the quantum networks modeling, which is responsible for the associations between input and output patterns in the proposed architecture (Svitek, 2008). The quantum entanglement mechanism can be well brought in neural networks modeling that provides more adaptability for parameter learning of the networks. Many researchers confirm that the entanglement mechanism is necessary to realize high-level quantum computing (Jozsa & Linden, 2003; Menneer, 1999), including quantum neural computing (QNC).

^{*} Corresponding author at: School of Reliability and Systems Engineering, Beihang University, Beijing, China.

E-mail addresses: yiqiancui@163.com (Y. Cui), shisjy@hotmail.com (J. Shi), wangzili2014@yahoo.com (Z. Wang).

<http://dx.doi.org/10.1016/j.neunet.2015.07.013>

0893-6080/© 2015 Elsevier Ltd. All rights reserved.

In existing QNN models, QNC is generally based on the real assumption, i.e. the quantum probability amplitudes are all real. Notwithstanding, the degree of quantum entanglement is limited if QNC is only defined in the real number domains. An extension from the real number domain to the complex number domain not only complies with more of the real physical world, but also can lead to a higher level of quantum entanglement by combined interaction of the real parts and the imaginary parts. In this way, the data information is encoded into an enhanced level of uncertainty, and the neural networks have a higher flexibility in parameter learning.

To improve the approximation and generalization ability of ANN by utilizing the mechanism of deep quantum entanglement within complex number domain, in this paper, a novel hybrid network model Complex Rotation Quantum Dynamic Neural Networks (CRQDNN) is proposed. The CRQDNN model is established on the definition of a new quantum neuron model based on the newly defined Quantum Complex Rotation Gate (QCRG). Since there are two freedom degrees considering the combination of the real and imaginary parts, a CQN has two output ports with each representing an individual portion of the deep quantum entanglement.

In this paper, the structure of the CRQDNN model is proposed, and the input/output mathematical relationships are derived. We select the Levenberg–Marquardt (LM) networks learning algorithm to ensure a fast convergence and a high probability of finding the global minimum. Two application studies are done in this paper, including the chaotic time series prediction and electronic remaining useful life (RUL) prediction. The experimental results show that CRQDNN is superior to traditional QNN model in the metrics of prediction accuracy.

The remainder of this paper is organized as follows. Section 2 sorts out the related works in previous researches. Section 3 introduces the background knowledge of qubits and quantum gates. Section 4 proposes the novel quantum neuron model CQR. Section 5 proposes the networks structure of CRQDNN with its input/output relationships. Section 6 conducts the application study in chaotic time series prediction. Section 7 conducts the application study in electronic RUL prediction. Finally, Section 8 concludes the whole paper.

2. Related works

2.1. DNN

DNN with different time-scales can model the dynamics of the short-term memory of neural activity levels (Lou & Cui, 2007; Meyer-Bäse et al., 1996). Recurrent Neural Networks (RNN) are a basic kind of DNN with feedback paths introducing dynamics into the model. Recurrent Neural Networks (RNN) are a closed loop system, with feedback paths introducing dynamics into the model. Unlike feed-forward neural networks, RNN can use their internal memory to process arbitrary sequences of inputs (Mandic & Chambers, 2001). Typical RNN models include fully recurrent network, Hopfield neural networks (HNN) (Hopfield, 1982), Elman neural networks (ENN) (Elman, 1990), Bi-directional RNN (Graves & Schmidhuber, 2005; Schuster & Paliwal, 1997), etc. However, these recurrent models merely utilize only the last one step of the feedback data. As a result, the memory capability does not always meet the requirement to treat time series with time accumulation effect.

To better deal with the problem, new kinds of dynamic neural networks are proposed that use the embedded filters as the memory units. Ayoubi (1994) presents a class of dynamic neuron model in which a filter is employed to generate the dynamics between the input and output of the system. A dynamic neuron model is constructed by adding an IIR filter within the standard static neuron structure. Compared with the one-way

DNN, the DNN with recurrent properties have a higher stability and robustness towards time series variances or noise. In this way, Back and Tsoi (1991) propose the dynamic neuron model with both IIR and finite impulse response (FIR) synapses, where the embedded FIR filter serves as the multi-step recurrent model. This kind of networks model maintains a short-term memory based on the mechanism of internal local feedback provided by the FIR filter. Yazdizadeh and Khorasani (2002); Frasconi, Gori, and Soda (1992) present another two similar forms of dynamic neuron model where the filter was placed after the activation function of the neuron.

A common feature of such kinds of DNN is, the interconnected dynamic neurons or the embedded filtering units contribute to the memory function. The memory parameters are adjustable according to specific requirements, which facilitate sequence or time series processing and prediction. However, a limitation of the existing DNN models is that they only process the time series data in a linear manner. There are some other kinds of DNN with their own dynamic designs, including Dynamic Wave Expansion Neural Networks (DWENN) (Lebedev, Steil, & Ritter, 2005), Dynamic Evolving Spiking Neural Networks (DESNN) (Kasabov, Dhoble, Nuntalid, & Indiveri, 2013), etc.

2.2. QNN

In recent years, quantum computing (QC) has attracted great attention worldwide because of its novel computational characteristics and potential computational efficiency. The concept of quantum computing is first stated based on the quantum mechanical nature of physical reality (Benioff, 1982; Feynman, 1982). Considerable interest has been generated in QC since Shor shows that large numbers can be factored in polynomial time (Shor, 1997) and Grover invents the searching algorithm that achieves quadratic speedup over classical searching algorithms (Grover, 1997). The quantum inspired computational intelligence (QCI) applications are enhanced based on QC principles. QCI has been shown to offer good performance in solving a wide range of problems and many efficient strategies have been proposed as improvement (Manju & Nigam, 2014). The most established research areas of QCI would include quantum evolutionary algorithms (Han & Kim, 2002; Kuk-Hyun & Jong-Hwan, 2004; Platel, Schliebs, & Kasabov, 2009), quantum fuzzy systems (Rigatos & Tzafestas, 2002; Seising, 2006), and QNN.

QNN is seen as the quantum analog of ANN, which is firstly proposed by Kak (1995). Lagaris, Likas, and Fotiadis (1997) innovatively propose the artificial quantum neuron model based on the quantum dynamics. Purushothaman and Karayiannis (1997) propose the model of quantum neural networks with multilevel hidden neurons based on the superposition of quantum states in the quantum theory. Kouda, Matsui, and Nishimura (2002); Matsui, Nishimura, and Peper (2005) propose the single layer quantum perceptron model for networks formulation. Zak and Williams (1998) realize the reconciliation of the linear reversible structure of quantum evolution with nonlinear irreversible dynamics of neural networks. Liu, Chen, Chang, and Shih (2013) propose a single-hidden-layer feed-forward quantum neural networks structure based on Grover quantum learning method.

The quantum neural networks can be modified by using different kinds of quantum gates or base state identifiers. To name a few, Matsui, Takai, and Nishimura (2000) invent a quantum neural networks model using the single bit rotation gate and two-bit controlled-not gate. Li, Li, Xiong, Chai, and Zhang (2014) propose the quantized neural networks model with the application of short-term electrical load forecasting, in which the quantum CNOT gate is embedded in the input layer quantum neurons. Further, Shafee (2007) proposes a completely different kind of networks

from the mainstream works, where the neurons are states connected by quantum gates.

Quantum entanglement is involved in the quantum networks modeling, which is responsible for the associations between input and output patterns in the proposed architecture (Svitek, 2008). In quantum physics, quantum entanglement refers to the situation where the quantum states of two or more objects have to be described with reference to each other, even though the individual objects may be spatially separated. In information sciences, quantum entanglement refers to the non-factorizable superposition of the basic quantum states (Ezhov & Ventura, 2000). It encompasses the intermediate states with interactions of basic states, and provides more flexibility than traditional computational manners in the training phase of the neural networks. It is demonstrated that the QNN models with quantum entanglement have at least no worse information processing capability than traditional networks (Kouda, Matsui, Nishimura, & Peper, 2005; Menneer & Narayanan, 1995).

The neural networks models above are all static models and they do not consider the time accumulation effect. To better model the temporal features, the quantum spiking NN model and a quantum computational neurogenetic model are discussed by N. Kasabov (2010); N.K. Kasabov (2007) by utilizing spatial-temporal information. Also, in an attempt to incorporate the time accumulation effect to model and predict time series behaviors, Li and Xiao (2014), Li et al. (2013)) propose two novel quantum behaved neural networks: Controlled-Hadamard Quantum Neural Networks (CHQNN) and Quantum-inspired Neural Networks with Sequence Input (QNN SI), which use temporal sequences or time series as the inputs.

In most of the existing QNN models, the quantum neural computing (QNC) is generally based on the real assumption, i.e. the quantum probability amplitudes are all real. This is also true for the existing quantum rotation gate only helps to shift the phase of the real angles. However, as is well known in quantum mechanics or quantum computing, a qubit is a two-level quantum system, described by a two-dimensional complex Hilbert space. The probability amplitudes of a qubit can necessarily be complex instead of real, and the quantum rotation can also be worked on the imaginary part of the qubits (as long as it is a unitary operation).

We address the issue above mainly because of the consideration that the real and imaginary parts can convey different information, and in this way a deeper quantum entanglement can be realized. In quantum neuron models, the quantum entanglement is only confined in the real number domain. Notwithstanding, a combined interaction of the real parts and the imaginary parts enables a deeper quantum entanglement. The data information is asymmetrically distributed to both the real parts and the imaginary parts, and the complex quantum rotation operation enhances the intrinsic uncertainty (or fuzziness) level. In this way, the data information is thus encoded into a higher degree of entanglement, and by combining such intrinsic fuzziness, the neural networks have a higher flexibility in parameter learning.

3. Qubits and quantum gates

In this section, we introduce the basic knowledge of QC that lays the foundation of the subsequent discussions, including the definition of the qubits and the quantum gates.

3.1. Qubits

In quantum computing, a qubit is a two-level quantum system, described by a two-dimensional complex Hilbert space. From the superposition principles, any state of the qubit may be written as

$$|\varphi\rangle = \cos\theta|0\rangle + e^{i\phi}\sin\theta|1\rangle. \quad (1)$$

Unlike the classical bit, which can only be set equal to 0 or 1, the qubit resides in a vector space parameterized by the continuous variables θ and ϕ . In Eq. (1), $\cos\theta$ and $e^{i\phi}\sin\theta$ are termed as the probability amplitudes (PA) of basis states $|0\rangle$ and $|1\rangle$. It is clearly seen that $|\cos\theta|^2 + |e^{i\phi}\sin\theta|^2 = \cos^2\theta + e^{i\phi}e^{-i\phi}\sin^2\theta = 1$, which satisfies the normalization condition called the probability axiom.

The Bloch sphere representation is useful to describe qubits, which provides a geometric picture of the qubit and of the transformations that one can operate on the state of a qubit. The Bloch sphere is shown in Fig. 1(a). Owing to the normalization condition, the qubit's state can be represented by a point on a sphere of unit radius, called the Bloch Sphere. This sphere can be embedded in a three-dimensional space of Cartesian coordinates.

For convenience, many researches confine their scope within the real domain. Their qubit states are represented by a point on a circle of unit radius as shown in Fig. 1(b), where the amplitudes of the basis states take real values. It can be written in the form as

$$|\zeta\rangle = \cos\gamma|0\rangle + \sin\gamma|1\rangle. \quad (2)$$

In this setting, $\cos\gamma$ and $\sin\gamma$ are termed as the probability amplitudes of the basis states $|0\rangle$ and $|1\rangle$, respectively. According to quantum theory, the probability measure of state $|0\rangle$ equals $\cos^2\gamma$, while the probability measure of state $|1\rangle$ equals $\sin^2\gamma$; we have $\cos^2\gamma + \sin^2\gamma = 1$, which satisfies the probability axiom.

3.2. Quantum Rotation Gate (QRG)

In quantum theory, the transformation of states is performed by the quantum transformation matrix. Such matrix is called a quantum gate, which forms the basic unit of quantum algorithms. The difference between the classical and quantum context is that a quantum gate has to be implemented reversibly and, in particular, must be a unitary operation. The definition QRG is given by

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \quad (3)$$

The QRG $R(\theta)$ shifts the phase of a qubit by θ . For the initial quantum state $|\zeta\rangle = [\cos\theta_0, \sin\theta_0]^T$, the function of the quantum rotation gate is given by

$$R(\theta)|\zeta\rangle = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta_0 \\ \sin\theta_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_0 + \theta) \\ \sin(\theta_0 + \theta) \end{bmatrix}. \quad (4)$$

3.3. Quantum Complex Rotation Gate (QCRG)

Since the quantum probability amplitude can necessarily be complex, here we propose a new quantum gate QCRG. Compared with QRG, the QCRG operation can exert an effect on the imaginary part of a qubit. The definition of QCRG is given by

$$CR(\theta, \phi) = \begin{bmatrix} e^{i\phi}\cos\theta & -e^{i\phi}\sin\theta \\ e^{i\phi}\sin\theta & e^{i\phi}\cos\theta \end{bmatrix} = e^{i\phi}R(\theta). \quad (5)$$

QCRG is a unitary operation, which is given by

$$\begin{aligned} CR(\theta, \phi) \cdot CR(\theta, \phi)^* &= \begin{bmatrix} e^{i\phi}\cos\theta & -e^{i\phi}\sin\theta \\ e^{i\phi}\sin\theta & e^{i\phi}\cos\theta \end{bmatrix} \begin{bmatrix} e^{-i\phi}\cos\theta & e^{-i\phi}\sin\theta \\ -e^{-i\phi}\sin\theta & e^{-i\phi}\cos\theta \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}_{2 \times 2}. \end{aligned}$$

Functioning QCRG on the initial quantum state $|\varphi\rangle = \cos\theta_0|0\rangle + e^{i\phi_0}\sin\theta_0|1\rangle$, we have

$$\begin{aligned} CR(\theta, \phi)|\varphi\rangle &= \begin{bmatrix} e^{i\phi}\cos\theta & -e^{i\phi}\sin\theta \\ e^{i\phi}\sin\theta & e^{i\phi}\cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta_0 \\ e^{i\phi_0}\sin\theta_0 \end{bmatrix} \\ &= \begin{bmatrix} e^{i(\phi_0+\phi)}\cos(\theta_0+\theta) \\ e^{i(\phi_0+\phi)}\sin(\theta_0+\theta) \end{bmatrix}. \end{aligned} \quad (6)$$

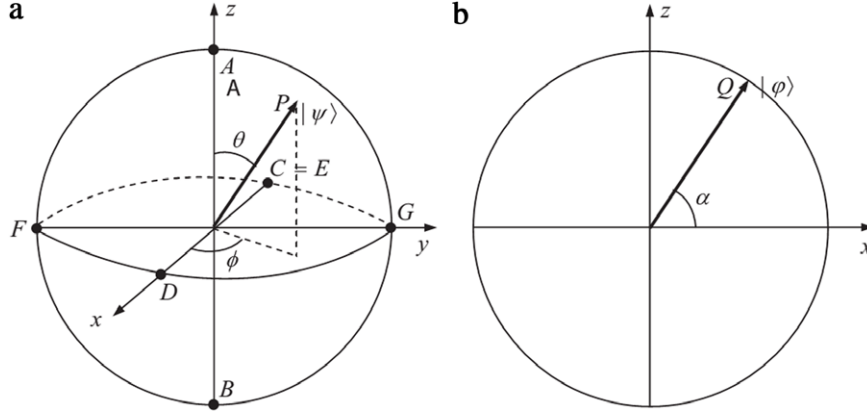


Fig. 1. Graphical qubit description. (a) Bloch sphere. (b) Qubit on unit circle.

With $CR(\theta, \phi)$, both the exponential part and the sinusoidal parts have the phase shifts by ϕ and θ , respectively.

4. Quantum neuron model

The quantum neuron model would be the most important part of the quantum neural networks model. Kouda et al. (2002) and Matsui et al. (2005) propose the single layer quantum perceptron model for the networks formulation, which is widely used as the quantum neuron model in the literature. The basic form of quantum neuron model proposed by Kouda et al. is shown in Fig. 2. This quantum neuron model only processes the real input quantized data $|x_h\rangle$ ($h = 1, \dots, n$), and utilizes the real quantum rotation gate as mentioned in Section 3.

Kouda's quantum neuron model only processes the real input quantized data $|x_h\rangle$ ($h = 1, \dots, n$), and utilizes the real quantum rotation gate as mentioned in Section 3. According to Fig. 2, the quantum neuron model involves the operations of phase shift, summation, reversal. The phase shift operation is done by using the quantum rotation gate, which is given by

$$R(\theta_h) = \begin{bmatrix} \cos \theta_h & -\sin \theta_h \\ \sin \theta_h & \cos \theta_h \end{bmatrix}.$$

The summation operation is done by $\sum_{h=1}^n R(\theta_h)|x_h\rangle$. Let α be the angle of the summation, i.e. $\alpha = \arg(\sum_{h=1}^n R(\theta_h)|x_h\rangle)$. The quantum reversal operation is done by the quantized CNOT gate $U(\alpha, \beta)$. The operational logic of $U(\alpha, \beta)$ is given by

$$U(\alpha_i, \beta) = \begin{bmatrix} \cos\left(\frac{\pi}{2}f(\beta) - \alpha\right) & -\sin\left(\frac{\pi}{2}f(\beta) - \alpha\right) \\ \sin\left(\frac{\pi}{2}f(\beta) - \alpha\right) & \cos\left(\frac{\pi}{2}f(\beta) - \alpha\right) \end{bmatrix} \quad (7)$$

where $f(\cdot)$ is the activation function. The output of the quantum neuron is defined by the probability amplitude corresponding to the quantum basis $|1\rangle$, i.e.

$$u = \sin\left(\frac{\pi}{2}f(\beta) - \arg\left(\sum_{h=1}^n R(\theta_h)|x_h\rangle\right)\right). \quad (8)$$

As is discussed in Section 2, a deeper entanglement can be realized when the data are artificially quantized into a complex form than into a real form. Considering this, we propose a novel quantum neuron model CQN based on Kouda's neuron model, which is shown in Fig. 3.

In CQN, all the quantized inputs are complex qubits. We use QCRG instead of traditional quantum rotation gate to realize the

phase shift operation. Since a complex qubit has two freedom degree, in CQN, the treatment of the summated qubit $\sum_{h=1}^n CR(\theta_h, \phi_h)|x_h\rangle$ is diverged in two branches. The first one treats with the real portion, which is given by $\alpha = \arg(\text{Re}(\sum_{h=1}^n CR(\theta_h, \phi_h)|x_h\rangle))$; the second one treats with the imaginary portion, which is given by $\rho = \arg(\text{Im}(\sum_{h=1}^n CR(\theta_h, \phi_h)|x_h\rangle))$. The final outputs are the result of the reversal operations acted on both the real branch and the imaginary branch.

The output of the quantum neuron is defined by the probability amplitude corresponding to the quantum basis $|1\rangle$. We have

$$u_1 = \sin\left(\frac{\pi}{2}f(\beta) - \arg\left(\text{Re}\left(\sum_{h=1}^n CR(\theta_h, \phi_h)|x_h\rangle\right)\right)\right) \quad (9)$$

and

$$u_2 = \sin\left(\frac{\pi}{2}f(\sigma) - \arg\left(\text{Im}\left(\sum_{h=1}^n CR(\theta_h, \phi_h)|x_h\rangle\right)\right)\right). \quad (10)$$

5. Networks structure of CRQDNN

In this section we propose the CRQDNN structure combining both CQNs and classical neurons. The mathematical relationships in the CRQDNN models are also exploited.

5.1. CRQDNN structure

The CRQDNN model is a 3 layer neural networks structure, as is shown in Fig. 4. From the input to the output, the layers are termed as the input layer (also termed as the quantum layer), the hidden layer, and the output layer, etc. The input layer consists of quantum neurons, while the hidden and output layers consist of common neurons.

For CRQDNN, the inputs are quantized sequences. Let n_1 , n_2 , and n_3 be the number of outputs concerning the quantum, hidden, and output layers, respectively; in this way, the number of CQNs is $n_0 = n_1/2$. In Fig. 4, $\mathbf{w}^{(output)} \in \mathbb{R}^{n_3 \times (n_2+1)}$ denotes the connection weights in the output layer, and $\mathbf{w}^{(hidden)} \in \mathbb{R}^{n_2 \times (n_1+1)}$ denotes the connection weights in the hidden layer. $f(\cdot)$ is the activation function. In our study, the sigmoid function $f(x) = 1/(1 + e^{-x})$ is used as the activation function.

We embed an IIR filter to fulfill the sequential computation tasks, which endow the networks with the memory function. The IIR filter possesses a feedforward status that is attached to the output ports of the quantum neurons. The internal structures of the IIR filter structure is depicted in Fig. 4, where N and M are the lengths of the filters, a_T ($T = 1, \dots, N+1$) is the IIR filter parameters (see Fig. 5).

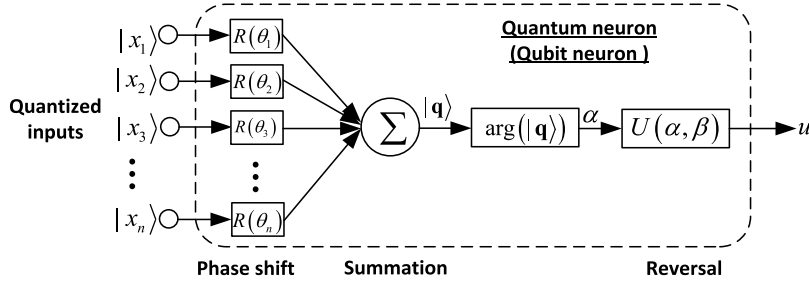


Fig. 2. Basic quantum neuron model.

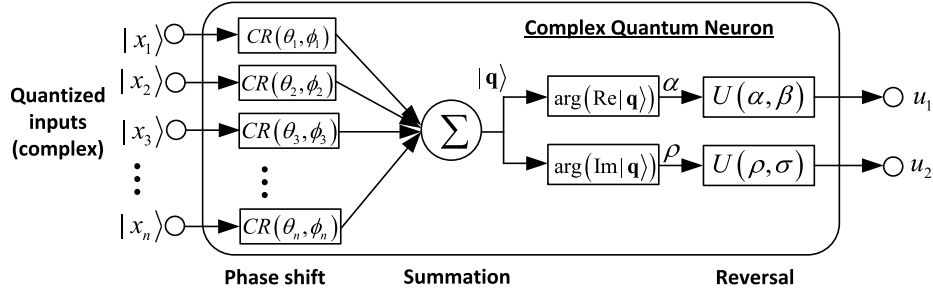


Fig. 3. CQN model.

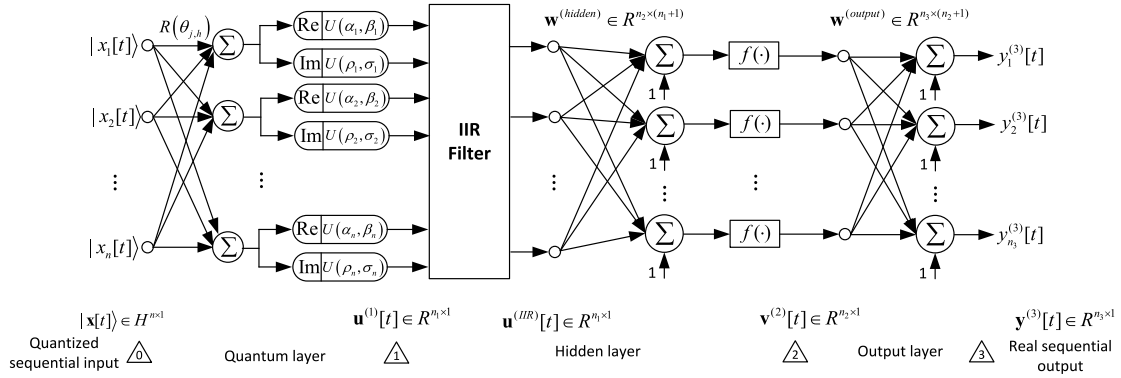


Fig. 4. CRQDNN structure.

5.2. Mathematical relationships

For the input layer, the laws of quantum physics give a description of the interactions between the qubit neurons and the classic neurons. The mathematical quantum descriptions of the real input vector $\mathbf{x}[t] \in R^{n \times 1}$ is given in the quantized form as

$$\begin{aligned} |x_h[t]\rangle &= \cos\left(\frac{2\pi}{1 + \exp(-x_h[t])}\right) |0\rangle \\ &+ \exp(i \arctan(x_h[t])) \sin\left(\frac{2\pi}{1 + \exp(-x_h[t])}\right) |1\rangle \\ &= \cos(\varphi_h[t]) |0\rangle + \exp(i\psi_h[t]) \cdot \sin(\varphi_h[t]) |1\rangle \end{aligned} \quad (11)$$

where $[t]$ represents the present time point, φ_h and ψ_h are the quantum phases of the h th element of the quantized input. When the quantum inputs are presented to quantized neural model, the quantum phase shift operation successively functions in qubit neurons. The quantum phase shift operation is given by

$$CR(\theta, \mathbf{f})|\mathbf{x}[t]\rangle = \begin{bmatrix} CR(\theta_{1,1}, \phi_{1,1}) & \cdots & CR(\theta_{1,n}, \phi_{1,n}) \\ \vdots & \ddots & \vdots \\ CR(\theta_{n_0,1}, \phi_{n_0,1}) & \cdots & CR(\theta_{n_0,n}, \phi_{n_0,n}) \end{bmatrix}$$

$$\begin{aligned} &\times \begin{bmatrix} |x_1[t]\rangle \\ \vdots \\ |x_n[t]\rangle \end{bmatrix} \\ &= [|q_1[t]\rangle \cdots |q_{n_0}[t]\rangle]^T \end{aligned} \quad (12)$$

with

$$|q_j[t]\rangle = \begin{bmatrix} \sum_{h=1}^n \cos(\varphi_h[t] + \theta_{j,h}) \\ \sum_{h=1}^n \exp(i\psi_h[t]) \cdot \sin(\varphi_h[t] + \theta_{j,h}) \end{bmatrix} \quad (13)$$

where $\theta \in R^{n_0 \times n}$ and $\mathbf{f} \in R^{n_0 \times n}$ represent the quantum rotation parameter matrices, and $CR(\theta, \mathbf{f})$ represents QCN, $|q_j[t]\rangle (j = 1, \dots, n_0)$ represents the quantum activation potential at time point t . In Fig. 4, $\alpha_j[t]$ and $\rho_j[t]$ represent the real and imaginary angles of $|q_j[t]\rangle (j = 1, \dots, n_0)$. From Eqs. (12) and (13), we have

$$\alpha_j[t] = \arctan \left(\operatorname{Re} \left(\frac{\sum_{h=1}^n \exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h})}{\sum_{h=1}^n \exp(i\phi_{j,h}) \cdot \cos(\varphi_h[t] + \theta_{j,h})} \right) \right). \quad (14)$$

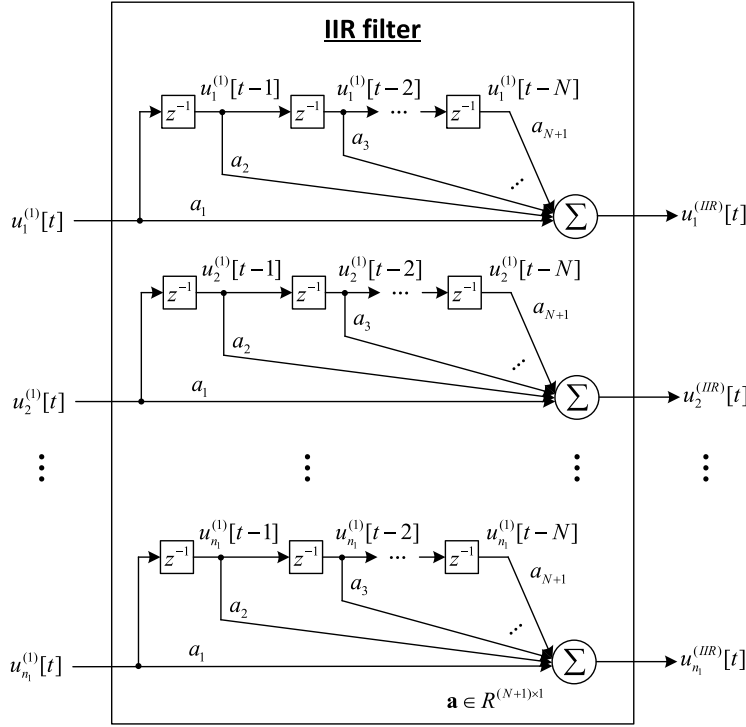


Fig. 5. Internal structures of the embedded IIR filter.

And

$$\rho_j[t] = \arctan \left(\operatorname{Im} \left(\frac{\sum_{h=1}^n \exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h})}{\sum_{h=1}^n \exp(i\phi_{j,h}) \cdot \cos(\varphi_h[t] + \theta_{j,h})} \right) \right). \quad (15)$$

The quantum reversal operation is given by

$$U(\alpha[t], \beta) = \begin{bmatrix} \cos\left(\frac{\pi}{2}f(\beta) - \alpha[t]\right) \\ \sin\left(\frac{\pi}{2}f(\beta) - \alpha[t]\right) \end{bmatrix} \quad (16)$$

and

$$U(\rho[t], \sigma) = \begin{bmatrix} \cos\left(\frac{\pi}{2}f(\sigma) - \rho[t]\right) \\ \sin\left(\frac{\pi}{2}f(\sigma) - \rho[t]\right) \end{bmatrix} \quad (17)$$

where $\beta \in R^{n_0 \times 1}$ represents the quantum reversal parameter vector corresponding to the real parts, and $\sigma \in R^{n_0 \times 1}$ represents the quantum reversal parameter vector corresponding to the imaginary parts. In this study, we use the probability of the base state $|1\rangle$ as the output, instead of the probability amplitude. Thus, the j th output of the quantum layer is given by

$$u_j^{(1)}[t] = \begin{cases} \sin^2\left(\frac{\pi}{2}f(\beta_j) - \alpha_j[t]\right), & j = 1, 3, \dots, n_1 - 1 \\ \sin^2\left(\frac{\pi}{2}f(\sigma_j) - \rho_j[t]\right), & j = 2, 4, \dots, n_1. \end{cases} \quad (18)$$

Let $u_j^{(IIR)}$ denote the output of the IIR filter connected to $u_j^{(1)}$; thus, it is calculated by

$$u_j^{(IIR)}[t] = \sum_{T=1}^{N+1} a_T u_j^{(1)}[t - T + 1], \quad j = 1, \dots, n_1. \quad (19)$$

Let $v_k^{(2)}$ denote the k th output of the hidden layer. Let $u_{n_1+1}^{(IIR)}[t] \equiv 1$, which serves as an external bias term of the hidden layer. The hidden layer outputs are thus calculated by

$$v_k^{(2)}[t] = f \left(\sum_{j=1}^{n_1+1} w_{k,j}^{(hidden)} f(u_j^{(IIR)}[t]) \right), \quad k = 1, \dots, n_2. \quad (20)$$

Let $v_{n_2+1}^{(2)}[t] \equiv 1$, which serves as an external bias term of the output layer. The output layer vector is defined by $\mathbf{y}[t] = [y_1[t], \dots, y_{n_3}[t]]^T$. The output values are thus calculated by

$$y_l^{(3)}[t] = \sum_{k=1}^{n_2+1} w_{l,k}^{(output)} v_k^{(2)}[t], \quad l = 1, \dots, n_3. \quad (21)$$

5.3. Networks learning

In CRQDNN, the adjustable parameters include the rotation angles of the quantum rotation parameters θ and \mathbf{f} , the quantum reversal vector β and σ , the IIR filter parameter vector \mathbf{a} , the weights in the hidden layer $\mathbf{w}^{(hidden)}$, and the weights in the output layer $\mathbf{w}^{(output)}$. To guarantee a fast convergence speed, we employ the Levenberg–Marquardt (LM) algorithm (Marquardt, 1963) to adjust the CRQDNN parameters. LM algorithm is a nonlinear optimization approach that lies between the Gauss–Newton method and the gradient descent method, for which the performance index is the mean squared error. It greatly reduces the chances that the evaluation function is trapped in local minima. Also, it is insensitive to overfitting, and it is effective to tackle with the problem of redundant parameters. Using LM algorithm, the networks would converge very fast in a high probability with the evaluation approximates the global minimum.

The fulfillment of LM algorithm depends on the gradient calculations based on the CRQDNN structures. From the output layer to the quantum layer, we obtain the gradient calculations in a back propagation scheme, which is discussed in detail in the Appendix.

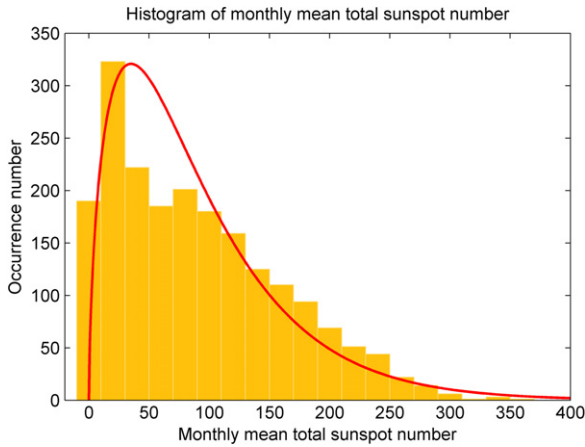


Fig. 6. Histogram of monthly mean total Sunspot number from Nov 1834 to Jun 2001.

6. Sunspot time series prediction application

In this section, we present the first application study towards chaotic time series prediction using CRDQNN. The famous Sunspot time series (SIDC, 2015) as the real-world experimental data is used as the experimental data.

6.1. Dataset description and preliminary statistical analysis

The Sunspot time series is an excellent indication of the solar activities for solar cycles, which impact the climate and weather patterns of the earth, as well as the satellite and space missions. It is well known that the approximate 11 year solar cycle actually consists of a 22 year magnetic cycle that flips polarity every 11 years.

The prediction of solar cycles is difficult due to its complexity. The yearly, monthly, as well as daily Sunspot time series can be obtained from the World Data Center for the Sunspot Index (SIDC). The Sunspot time series can be seen as chaotic systems with noise, which have the characteristics of aperiodic, bounded, deterministic, and sensitive to initial conditions (Abarbanel, 1996). Many prediction methods have been exploited to the related time series, and in most cases the frequently used predictors are ANN models (Marra & Morabito, 2006).

The truncated sub-series from November 1834 to June 2001 is selected that consists of 2,000 months. This interval is also selected to compare the performance of the proposed methods from the researches, e.g. (Ardalani-Farsa & Zolfaghari, 2010; Gholipour, Araabi, & Lucas, 2006). With respect to the data of the monthly mean total Sunspot number within this period, its mean and standard deviation values are calculated as 90.1824 and 70.5796, respectively. Fig. 6 provides the histogram of the monthly mean total Sunspot number, which shows a clear left-skewed pattern. The histogram can be represented using a two-parameter Gamma distribution given by

$$f(x) = \frac{x^{\alpha-1}}{\beta^{\alpha} \Gamma(\alpha)} \exp\left(-\frac{x}{\beta}\right) \quad (22)$$

where $\alpha > 0$ is the shape parameter, $\beta > 0$ is the scale parameter, and $\Gamma(\cdot)$ represents the Gamma function. The shape of the Gamma distribution corresponding to the Sunspot data is also shown in Fig. 6. From a statistical analysis, the shape parameter and scale parameter are estimated as $\hat{\alpha} = 1.6326$ and $\hat{\beta} = 55.2378$.

In this experiment, the Sunspot time series of 13-month smoothed monthly total Sunspot number is chosen as the data under investigation. The first 1000 samples are used for training, while the remaining 1000 samples are used for testing and validation.

6.2. Experimental configurations

For CRDQNN, the input time series is scaled in the range of $[-1, 1]$ to make sure that the initial phases of the quantum neurons are confined within the interval of $[0, 2\pi]$. Since we do not require the output time series to be normalized, we do not use any activation function for the output layer.

In our study, we use the Levenberg–Marquardt algorithm to facilitate a fast training of the CRDQNN model. The initial velocity parameter is selected as $\mu_0 = 0.001$, while the multiplying parameter is selected as $v = 10$. The maximum iteration number is 50, while the termination condition is that the weight position increment is lower than $\delta = 1e-5$.

The performance under different networks structures or configurations are different. In our study, we conduct 100 independent experiments for each networks configuration, and in each experiment the initial parameters of the networks models are randomly chosen. The performance results are shown in the sense of average values, best values, and standard deviations. Further, the CRDQNN performance is also compared with (1) Kouda's Qubit Neural Networks (Kouda et al., 2002) with IIR filter (QNN-IIR) and (2) traditional Elman networks.

6.3. Experimental results

To evaluate the prediction performance, the root mean squared error (RMSE) and normalized mean squared error (NMSE) are used to measure the prediction performance of the recurrent neural networks. These are given in the following equations:

$$RMSE = \sqrt{\frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} (\tilde{\mathbf{y}}[t] - \mathbf{y}[t])^T (\tilde{\mathbf{y}}[t] - \mathbf{y}[t])} \quad (23)$$

$$NMSE = \frac{\sum_{t=1}^{t_{\max}} (\tilde{\mathbf{y}}[t] - \mathbf{y}[t])^T (\tilde{\mathbf{y}}[t] - \mathbf{y}[t])}{\sum_{t=1}^{t_{\max}} (\tilde{\mathbf{y}}[t] - \bar{\mathbf{y}})^T (\tilde{\mathbf{y}}[t] - \bar{\mathbf{y}})} \quad (24)$$

where $\bar{\mathbf{y}}$ is the average value of the training data during the whole time span.

At the beginning, the number of quantum layer and the hidden layer is empirically evaluated. At this time, we fixed the length of IIR filter as $N = 5$, and the length of FIR filter as $M = 5$.

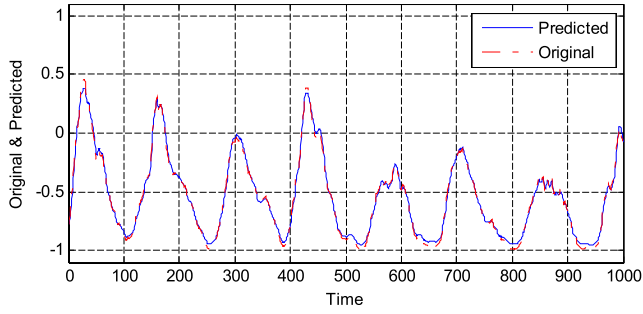
The experimental results calculated from 100 independent experiment runs are listed in Table 1 for the Sunspot datasets, including mean RMSE, best RMSE, mean NMSE, as well as the standard variation values.

In Table 1, the best mean RMSE results are highlighted in bold. As is seen in Table 1, the experimental results using CRDQNN are quite outstanding among all the 3 methods employed in this study. We can see that for each group divided by the hidden neuron number, the mean RMSE values for CRDQNN are largely about or lower than $2.0E-2$, and almost all of the best RMSE values are lower than $8.0E-3$. The experimental values under QNN-IIR are already satisfactory to our expectation, with the mean RMSE values are lower than $3.0E-2$, and the best RMSE values are lower than $1.0E-2$. However, the results under QNN-IIR are still inferior to the results under CRDQNN in most of the cases. Also, the RMSE results under CRDQNN are almost about $1/5 \sim 1/8$ of the values using the traditional Elman recurrent networks.

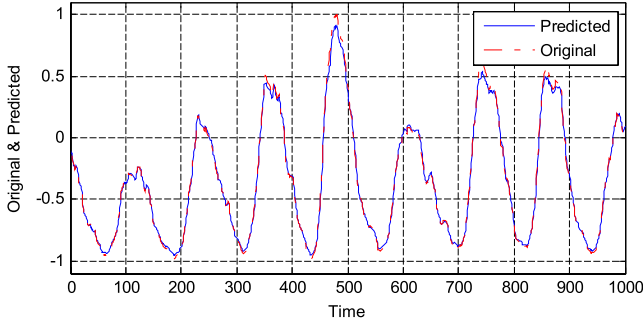
The similar conclusions can also be drawn considering the NMSE results. As is shown in Table 1, the NMSE values under CRDQNN are generally $5 \sim 10\%$ lower than those under QNN-IIR for almost every line. Together with the RMSE results, it is demonstrated that the CRDQNN model has a strong capability of time series prediction with sequential inputs.

Table 1
Prediction performance of CQDNN, QNN-IIR, Elman for the Sunspot time series.

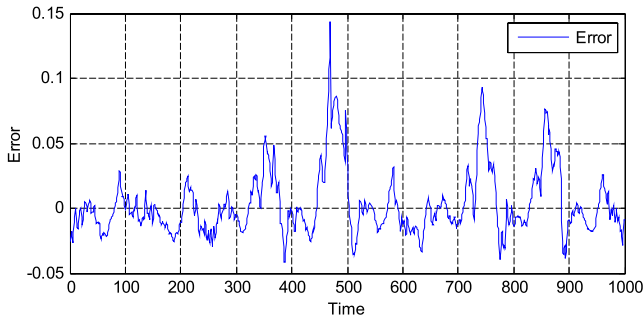
Hidden	Quantum	CQDNN				QNN-IIR				Elman			
		RMSE mean	RMSE best	NMSE mean	Std	RMSE mean	RMSE best	NMSE mean	Std	RMSE mean	RMSE best	NMSE mean	Std
4	4	2.14E-02	7.87E-03	3.07E-03	7.61E-03	2.33E-02	8.87E-03	3.85E-03	9.74E-03	9.61E-02	4.84E-02	9.40E-03	1.98E-02
	8	1.72E-02	5.99E-03	1.53E-03	7.19E-03	1.98E-02	6.91E-03	2.24E-03	6.99E-03				
	12	1.70E-02	5.66E-03	1.74E-03	4.82E-03	1.66E-02	5.75E-03	1.96E-03	6.69E-03				
	16	1.65E-02	5.32E-03	1.85E-03	5.59E-03	1.55E-02	5.05E-03	1.71E-03	6.75E-03				
8	4	1.84E-02	7.01E-03	1.60E-03	4.73E-03	2.13E-02	6.43E-03	2.04E-03	8.32E-03	7.31E-02	3.80E-02	6.76E-03	1.78E-02
	8	1.54E-02	6.55E-03	8.38E-04	3.87E-03	1.80E-02	7.01E-03	1.38E-03	6.48E-03				
	12	1.34E-02	5.65E-03	9.20E-04	5.45E-03	1.87E-02	5.69E-03	1.05E-03	5.07E-03				
	16	1.31E-02	4.76E-03	9.73E-04	4.46E-03	1.29E-02	5.32E-03	9.88E-04	4.69E-03	6.30E-02	2.55E-02	5.23E-03	1.65E-02
12	4	1.56E-02	7.01E-03	1.44E-03	3.26E-03	1.93E-02	7.22E-03	1.54E-03	7.68E-03				
	8	1.47E-02	5.18E-03	1.23E-03	5.61E-03	1.60E-02	6.76E-03	1.42E-03	5.36E-03				
	12	1.16E-02	5.22E-03	7.05E-04	3.60E-03	1.43E-02	5.45E-03	8.97E-04	4.66E-03				
	16	1.11E-02	4.89E-03	8.40E-04	3.30E-03	1.38E-02	4.77E-03	8.56E-04	4.71E-03				
16	4	1.56E-02	6.23E-03	1.11E-03	4.05E-03	1.88E-02	4.90E-03	1.74E-03	6.92E-03	5.67E-02	1.86E-02	5.38E-03	1.61E-02
	8	1.44E-02	4.98E-03	9.36E-04	3.96E-03	1.55E-02	5.75E-03	1.25E-03	3.86E-03				
	12	1.21E-02	4.64E-03	9.21E-04	2.63E-03	1.43E-02	5.03E-03	9.70E-04	4.44E-03				
	16	1.14E-02	3.94E-03	8.67E-04	4.84E-03	1.29E-02	4.68E-03	8.00E-04	4.94E-03				



(a) Performance on the training data.



(b) Performance on the test data.



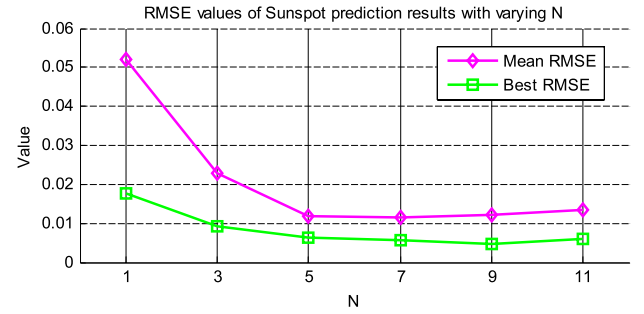
(c) Error on the test data.

Fig. 7. Typical prediction by CRQDNN for Sunspot time series.

For CRQDNN, the more the number of the quantum neurons, the better the prediction performance. This is consistent with our heuristic knowledge about quantum neural networks model that the more the number of quantum neurons, the higher the level of quantum entanglement. For CRQDNN, it utilizes not only the real portion quantum entanglement, but also the corresponding imaginary portion. The entanglement is more complicated than QNN-IIR, which provides more redundancy and flexibility in networks parameter learning. It is demonstrated that CRQDNN takes more of the advantage of quantum entanglement in the training process compared with QNN-IIR.

Fig. 7 shows that the CRQDNN model is trustworthy to give good prediction performance with low prediction error. It is able to tackle the noise in the Sunspot time series.

We also make a more detailed comparison of the chaotic time series prediction results with other methods published in the literature, see Table 2. From the perspective of NMSE, with respect to the Sunspot dataset, the prediction results of the proposed CRQDNN model are generally better than (or can be compared to) other methods apart from Hybrid NARX-Elman RNN with residual analysis (Ardalani-Farsa & Zolfaghari, 2010).

**Fig. 8.** RMSE performances comparison under different temporal structures.

6.4. Results under different temporal structures

Aside from the number of quantum layer or hidden layer neurons, the temporal structures in the CRQDNN recurrent networks model are also very important for the networks performances. The total delay timers of the IIR filter, also exert an effect on the time series prediction performances. In this part, we investigate the effect under different temporal structures. We obtain the RMSE values of the Sunspot time series prediction results corresponding to varying N , which are shown in Fig. 8. The mean and best RMSE curves all have a decreasing tendency when N increases, meaning that a high value of N would generally bring down the prediction errors. When N is small (e.g. $N = 1$), the RMSE value is very high, and the prediction accuracy is correspondingly low. At the same time, when N exceeds 5, the decrease of RMSE values becomes quite insignificant. Since a higher N means a more complex networks structure and higher computational efforts, $N = 5$ might be the best choice to facilitate an accurate prediction.

7. Electronic prognostics application

Addressing on the requirement of early fault prediction with high accuracy in EPHM, we focus on the issue of electronic prognostics, and show that the CRQDNN model can fulfill the RUL estimation task. In this section, we will show that CRQDNN has the ability to track the major deterioration trends of the developing fault, and it is also robust towards noise compared with methods based on traditional networks models.

7.1. Fundamentals of prognostics and RUL estimation

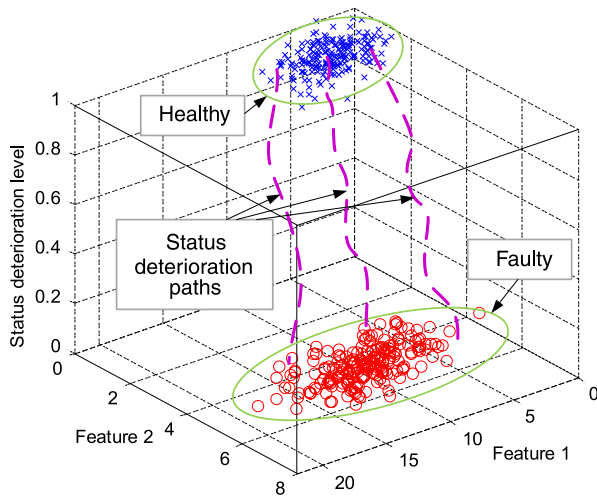
Electronic Prognostics and Health Management (Electronic PHM) is gaining more and more attention since it is an important and integral part of reliable and safe performances of electronic systems. It is an enabling discipline of technologies and methods with the potential of solving reliability and testability problems due to complexities in design, manufacturing, environmental and operational use conditions, and maintenance (Gu, Lau, & Pecht, 2009; Pecht & Jaai, 2010). The prevention of electronic faults during operation requires the prediction of the remaining useful life (RUL) based on deterioration or declination monitoring, especially for the critical circuit units or catastrophic faults (Sarathi Vasan, Long, & Pecht, 2013).

The RUL prediction relies on the current status identification of the monitoring electronic product. As is shown in Fig. 9, a product gradually turns from the healthy operational state to the faulty state, and the identified level of status deterioration (or fault development) serves as the prerequisite to make RUL prediction. The deterioration paths do not always follow the same pattern, and the effect of noise or inaccurate measurement even worsens the situation. The task in this section is to abstract the useful information hidden in the data and facilitate a reliable RUL prediction.

Table 2

A comparison with the results from the literature on the Sunspot time series.

Prediction method	RMSE	NMSE
McNish–Lincoln (Sello, 2001)		8.00E–02
Multi-layer perceptron/Elman RNN (Koskela et al., 1996)		9.79E–02
Wavelet packet multilayer perceptron (Teo et al., 2001)		1.25E–01
Radial basis networks with orthogonal least squares (RBF-OLS) (Gholipour et al., 2006)		4.60E–02
Locally linear model tree (LLNF-LoLiMot) (Gholipour et al., 2006)		3.20E–02
Hybrid NARX-Elman RNN with RA (Ardalani-Farsa & Zolfaghari, 2010)	1.19E–02	5.90E–04
Synapse Level Cooperative Coevolution RNN (SL-CCRNN) (Chandra & Zhang, 2012)	1.66E–02	1.47E–03
Neuron Level Cooperative Coevolution RNN (NL-CCRNN) (Chandra & Zhang, 2012)	2.60E–02	3.62E–03
Proposed CRQDNN	1.16E–02	7.05E–04

**Fig. 9.** Illustration of status deterioration.

The data-driven strategy is universally used in prognostics and RUL prediction since it can detect and utilize the useful information “hidden” in the monitoring or training data (Kabir, Bailey, Lu, & Stoyanov, 2012; Pecht & Jaai, 2010). The information of the past history to infer the future by updating the fault time prediction and provide RUL estimations. For data-driven methods, the prognostic effectiveness of the electronic products is highly dependent on the prediction accuracy of the prognostic model or method.

7.2. Experimental setup

The electronic circuit under test is the secondary power conversion board in the XD-3539A airborne radio frequency analyzing instrument. The power conversion board is expected to convert the input DC voltage to the standard +12 V DC voltage output which can be utilized by the downstream modules. A low dropout (LDO) regulator chip realizes the power conversion function. There

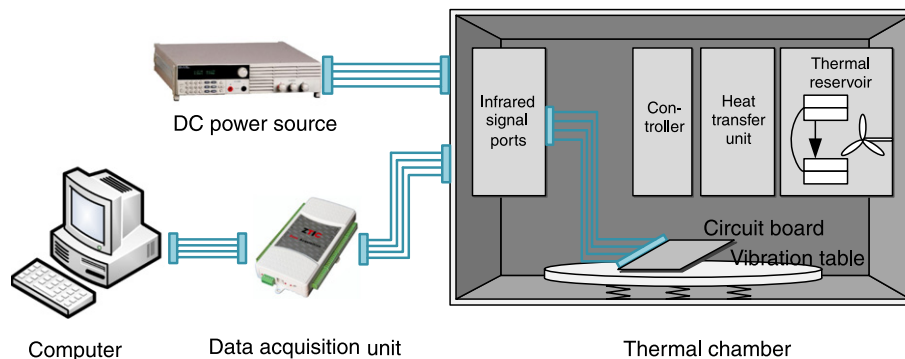
are also some small-scaled peripheral circuits responsible for the power control.

The power conversion board is very important for the whole instrument, therefore it is required to learn the state declination pattern for the purpose of fault identification as well as prognostics. Accelerated aging methodologies are integral to the induction of degradation and component faults into the electronic products. In an accelerated aging scheme, we conduct the disruptive experiment for a total of 4 identical power conversion board, and exert continuous thermal cycling on them on a high temperature level. We choose thermal cycling with vibration as the aging condition because thermal cycling is strongly associated with die solder degradation and wire lift, and vibration is closely associated with the wire or pin fractures. The experimental system schematic of the thermal cycling with vibration is illustrated in Fig. 10.

Accelerated aging methodologies are integral to the induction of degradation and component faults into the electronic products. In an accelerated aging scheme, we conduct the disruptive experiment for a total of 4 identical power conversion board at the same time, and exert continuous thermal cycling on them on a high temperature level. We choose thermal cycling because it is strongly associated with die solder degradation and wire lift.

In this study, the converted voltage output serves as the prognostic indicator revealing the declination level of the circuit board, which can be used by prognostics algorithms to estimate the RUL. The HL4005T temperature chamber is used to provide the high and low temperature environment. The voltage outputs for the circuits are induced and collected by the voltage sensors placed outside the chamber. The data are sampled and sent through a data acquisition device to a computer running a LabView interface. The default sampling time is 10 s.

There is a relatively long period of perfect operation during the accelerated aging experiment, after which the fault declination begins to be observed. In this study, we only take the last thousands of samplings representing the declination stage to make the following analyses.

**Fig. 10.** Overview of the accelerated aging experimental system.

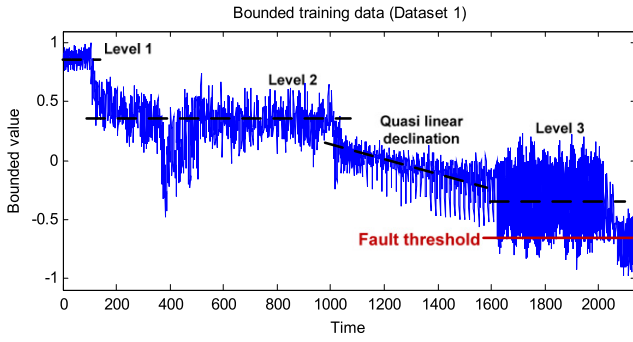


Fig. 11. Monitoring output voltage values of the training data.

7.3. Data analysis and training process

We obtain 4 datasets of output voltage corresponding to the 4 power conversion circuit boards in the period of accelerated aging experiment. The data are all bounded within the interval of $[-1, 1]$. All the 4 datasets reveal the deterioration process from the normal operational state to the faulty state. The deterioration process is not following a monotonous pattern; instead, it involves significant fluctuations and repetitions. Additionally, there is heavy sensor noise on the monitoring signals. Of course, we do not know exactly what internal changes lead to such fault deterioration pattern, and the data-driven prognostic method would be the reasonable choice to solve the problem.

For the 4 experimental power conversion boards, the degrees of fluctuation and repetition vary from each other, as well as the actual time to fault (TTF) values. The observed TTF values for them are 2059, 2032, 2294, 2107 time units, respectively. We select Dataset 1 as the training data, and the rest datasets are used for model testing. Fig. 11 depicts the output voltage values over time. From Fig. 11 we can see that the output voltage signals are not continuously decreasing; instead, there are several “horizontal” phases with the voltage values maintaining on the same level, which are marked with the dash lines in Fig. 11.

We use the CRQDNN model to make the RUL prediction with the voltage value bounded in $[-1, 1]$. The bounded values are also quantized to fulfill the requirement of the quantum neurons. We take the training data (i.e. Dataset 1) as an example to make the

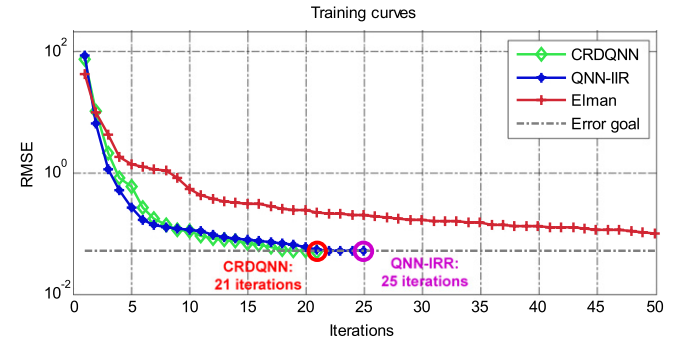


Fig. 13. Training curves for CRQDNN and Elman networks.

quantization. The real and imaginary parts of the quantum amplitude values in the basis state of $|0\rangle$ and $|1\rangle$ are calculated, which is shown in Fig. 12. Fig. 12 depicts the deep entanglement with the complex quantum rotation gate. The qubits after quantization using complex quantum rotation gate convey more information than the qubits using traditional quantum rotation gate.

Note that we utilize the filtering function of CRQDNN, and do not denoise the data when using CRQDNN to predict RUL. Based on the LM networks learning algorithm, we obtain the trained CRQDNN model. The number of quantum and hidden neurons are both 10. Besides, the lengths of IIR and FIR filters are also chosen to be 10.

In order to make a comparison, we also use the QNN-IIR method and the traditional Elman neural networks model in the RUL prediction. In this setting, since the traditional Elman recurrent neural networks lack the capability to deal with data with heavy noise, we use the Daubechies wavelet (DB8) to denoise the training voltage signal with 8 levels of the decomposition components. We also use the Levenberg–Marquardt algorithm to train the Elman recurrent networks. The number of hidden neurons is 10.

Fig. 13 shows the training error curves of the methods. We use the RMSE as the training error metric, and the error goal is set to be 0.05. From Fig. 13, for CRQDNN training, only 21 iterations are required to make sure that the RMSE value turns below the error goal line. If we use QNN-IIR, 25 iterations are required to satisfy the convergent condition. By contrast, the Elman recurrent networks have not met the requirement in the first 50 iterations. Also, the

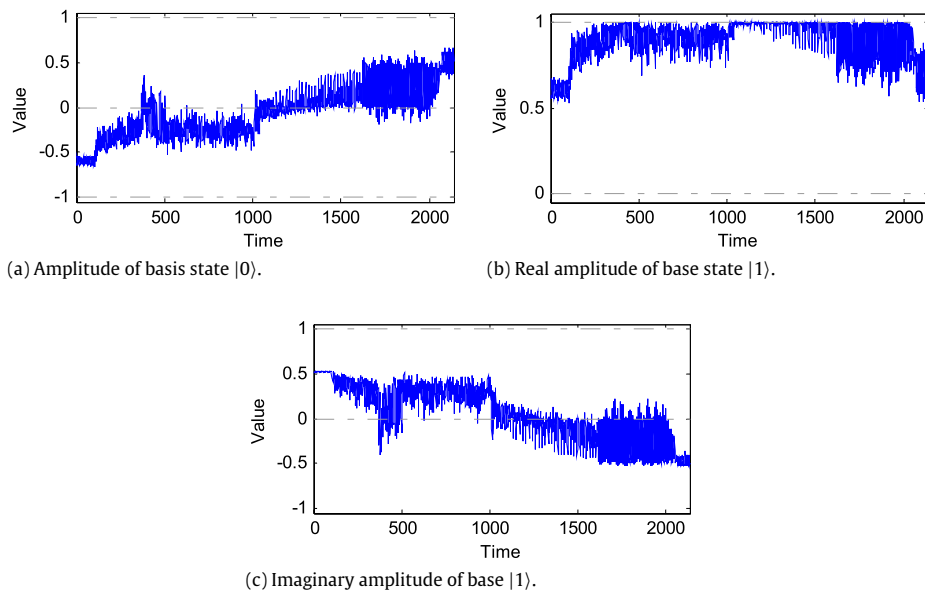


Fig. 12. Quantized amplification portions of basis states for the training data.

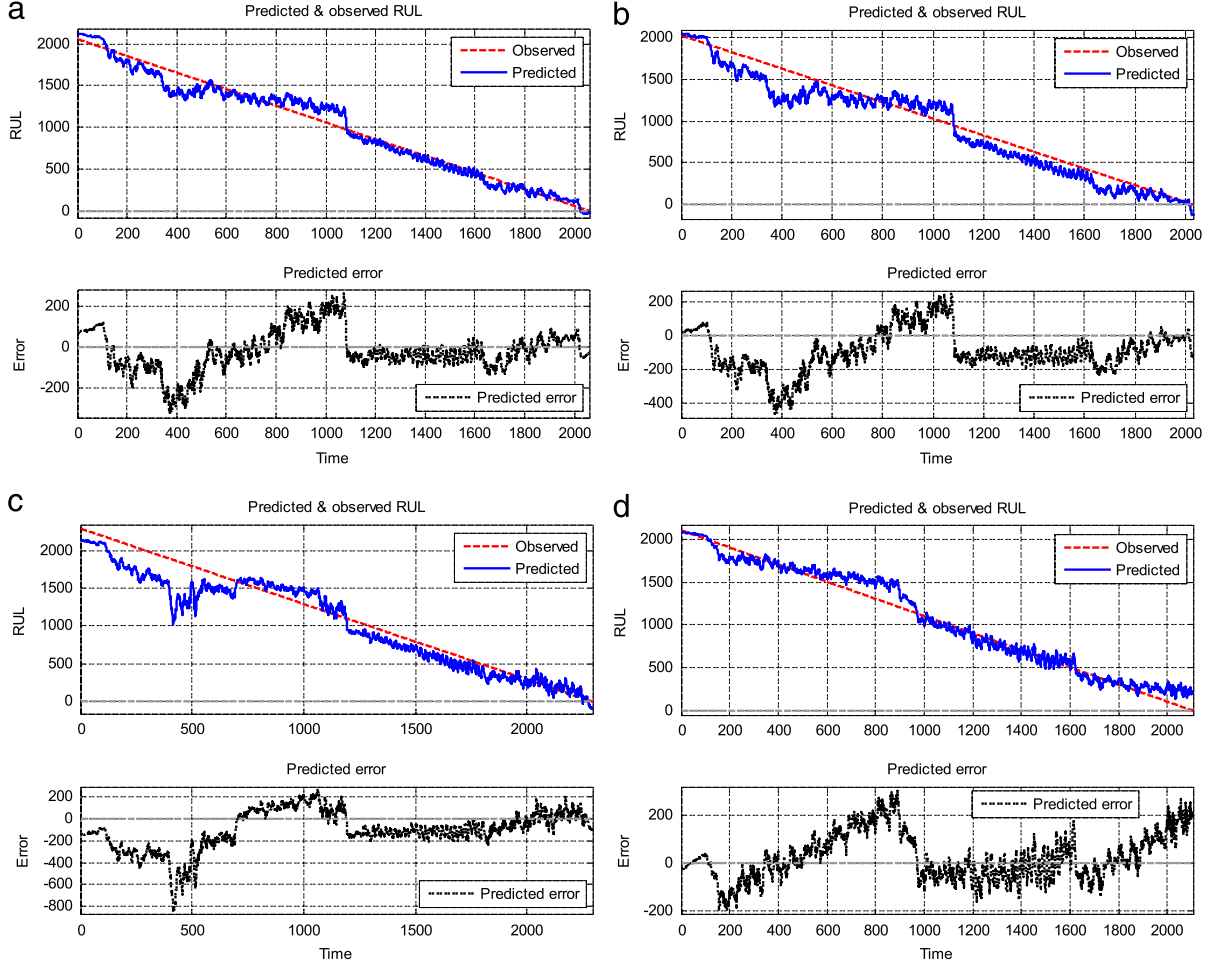


Fig. 14. Comparison with predicted and observed RUL. (a) Dataset 1 (training data), (b) Dataset 2, (c) Dataset 3, (d) Dataset 4.

final error value for Elman networks is 0.0834, while the value for CRQDNN is 0.0495. It manifests that the CRQDNN model has a comparatively more robust convergence capability.

7.4. Experimental results

We conduct 100 independent experiments for each networks configuration, and in each experiment, the initial parameters of the networks models are randomly chosen. We use the trained CRQDNN model to function on the training data as well as the 3 test datasets. The typical predicted system deterioration curves are shown in Fig. 14. The prediction curves track the real observed deterioration process well on a large scale, and the lifetime estimation results are quite reliable. This is especially true for the quasi linear declining stage (about 1050 ~ 1600 time points) where the fitness level is the highest during the whole period. By contrast, the prediction deviation degree is high when entering or leaving the horizontal phases.

We select RMSE, NMSE, as well as Mean absolute percentage error (MAPE) as the characterizing metrics to make a quantitative evaluation of the prediction accuracy. The mathematical forms of RMSE and NMSE have been presented in Section 6, and MAPE is given by the following equation:

$$MAPE = \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} \left\| \frac{\hat{y}[t] - y[t]}{y[t]} \right\|_2. \quad (25)$$

Table 3

Prediction error under CRQDNN for the datasets.

	MAPE	RMSE	NMSE
Dataset 1	0.1558	0.0655	0.0455
Dataset 2	0.2291	0.0935	0.0633
Dataset 3	0.2308	0.0903	0.0758
Dataset 4	0.1877	0.0652	0.0320

The mean MAPE, RMSE, and NMSE values of the 100 experimental runs are calculated and listed in Table 3. It is seen that with respect to the predicted RUL results (bounded to [0, 1]), the MAPE, RMSE, and NMSE values are all lower than 0.3, 0.1, and 0.1, respectively. Considering the fact that there are more than 2000 data points, the prediction effectiveness and accuracy are quite optimistic. The prediction accuracy for Dataset 3 is the lowest. This is because that the TTF value for Dataset 3 is about 2300 time units. Its changing pattern is also the most distinct one from the training data. By contrast, Dataset 4 is the best one among the three test datasets considering the prediction accuracy metrics.

Next, we make a comparison with other methods. We use both the original and denoised training data and test data as the input for CRQDNN, QNN-IIR, and traditional Elman recurrent networks. The final results are listed in Table 4.

Heuristically, both CRQDNN and QNN-IIR have the ability to process data with noise, as a result of the dynamic properties provided by its embedded IIR filters. With respect to CRQDNN, the differences of the prediction errors using the original data and the denoised data are not significant, which demonstrates that CRQDNN has a comparatively strong ability to “learn from

Table 4

Comparison of RUL prediction results using different methods.

Dataset	Method	MAPE	RMSE	NMSE
Dataset 3	CRQDNN with original data	0.2308	0.0903	0.0758
	QNN-IIR with original data	0.3182	0.1044	0.0887
	CRQDNN with denoised data	0.2192	0.0754	0.0631
	QNN-IIR with denoised data	0.1961	0.0632	0.0544
	Elman with denoised data	0.6161	0.1882	0.1448
Dataset 4	CRQDNN with original data	0.1877	0.0652	0.0320
	QNN-IIR with original data	0.2776	0.0871	0.0408
	CRQDNN with denoised data	0.2465	0.0910	0.0621
	QNN-IIR with denoised data	0.2541	0.1109	0.0530
	Elman with denoised data	0.4891	0.1450	0.0916

noise". It can also be seen in Table 4 that the prediction accuracy under CRQDNN is generally higher than QNN-IIR if we use the denoised data. This can be explained by the fact that CRQDNN takes advantage of a deeper quantum entanglement compared with QNN-IIR, and correspondingly the former can bring the filtering function into better play. It is interesting that the prediction results under CRQDNN are a little inferior to the results under QNN-IIR for the group of denoised data.

It is also seen in Table 4 that the prediction results using CRQDNN (with original data) are better than the Elman recurrent networks (with denoised data). The MAPE, RMSE, and NMSE values for CRQDNN are about $1/2 \sim 2/3$ less than the Elman networks.

8. Conclusions

To enhance the approximation and generalization ability of ANN by utilizing the mechanism of deep quantum entanglement within complex number domain, a novel hybrid networks model Complex Rotation Quantum Dynamic Neural Networks (CRQDNN) is proposed in this paper. CRQDNN is a three layer model with a quantum layer which employs the Quantum Complex Rotation Gate (QCRG) and quantum CNOT gate, and a hidden layer which employs the IIR filter as the memory units. Compared with previous real QNN model, the proposed complex QNN model enhances the level of quantum entanglement, i.e. the intrinsic uncertainty level. The data information is thus encoded into a higher degree of entanglement, and by combining such intrinsic fuzziness, the neural networks have a higher flexibility in parameter learning. Also, to handle time series input, an IIR filter is embedded in the hidden layer to endow the networks model with the memory capability and dynamic features.

Two application studies are done in this paper, including the chaotic time series prediction and RUL prediction under the task of electronic prognostics. In the first application study, the average RMSE and NMSE values under CRQDNN are generally 5%~10% lower than the values using traditional real QNN models. This identifies that the former has taken advantage of deep quantum entanglement created by the complex rotation method. The second application study confirms the prediction accuracy of CRQDNN. The experimental results demonstrate an outstanding prediction accuracy of the proposed CRQDNN model compared with traditional real QNN models. At the same time, the RUL prediction effectiveness using the original data can be compared to the cases using the denoised data, which shows that the CRQDNN model has a higher capability to deal with noise.

Acknowledgments

We express sincere appreciation to the editor and reviewers for their efforts to improve this paper. We are very thankful for the help from No. 10 Research Institute, Electronics Technology Group

of China. This work is supported by Science and Technology on Reliability and Environmental Engineering Laboratory, Beihang University. This work is also sponsored by fund projects 61316705, 51319040301 and Z132014B002.

Appendix. Gradient calculations of CRQDNN

For the output layer of CRQDNN, the increment of $\mathbf{w}^{(output)}$ is calculated by

$$\Delta w_{l,k}^{(output)} = \frac{\partial e[t]}{\partial y_l^{(3)}[t]} \frac{\partial y_l^{(3)}[t]}{\partial w_{k,l}^{(output)}} = y_k^{(2)}[t] \quad (l = 1, \dots, n_3, k = 1, \dots, n_2 + 1). \quad (26)$$

For the hidden layer, the increment of $\mathbf{w}^{(hidden)}$ is calculated by

$$\begin{aligned} \Delta w_{k,j}^{(hidden)} &= \frac{\partial e[t]}{\partial v_k^{(2)}[t]} \frac{\partial v_k^{(2)}[t]}{\partial w_{k,j}^{(hidden)}} \\ &= \sum_{l=1}^{n_3} \left(-\frac{\partial e[t]}{\partial y_l^{(3)}[t]} \frac{\partial y_l^{(3)}[t]}{\partial v_k^{(2)}[t]} \right) \frac{\partial v_k^{(2)}[t]}{\partial w_{k,j}^{(hidden)}} \\ &= g(v_k^{(2)}[t]) \sum_{l=1}^{n_3} w_{l,k}^{(output)} \cdot f(u_j^{(IIR)}[t]) \\ &= \delta_k^{(2)}[t] \cdot f(u_j^{(IIR)}[t]) \quad (k = 1, \dots, n_2) \end{aligned} \quad (27)$$

where $\delta_k^{(2)}[t] = g(v_k^{(2)}[t]) \sum_{l=1}^{n_3} w_{l,k}^{(output)}$, and $g(\cdot)$ is the first order derivation of the activation function given by $g(x) = \frac{d}{dx}f(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1-f(x)}{f(x)^2}$.

The increment of the IIR filter parameter vector \mathbf{a} is calculated by

$$\begin{aligned} \Delta a_T &= \sum_{j=1}^{n_1} \frac{\partial e[t]}{\partial u_j^{(IIR)}[t]} \frac{\partial u_j^{(IIR)}[t]}{\partial a_T} \\ &= \sum_{j=1}^{n_1+1} \sum_{k=1}^{n_2} \frac{\partial e[t]}{\partial v_k^{(2)}[t]} \frac{\partial v_k^{(2)}[t]}{\partial u_j^{(IIR)}[t]} \frac{\partial u_j^{(IIR)}[t]}{\partial a_T} \\ &= \sum_{j=1}^{n_1+1} \left(g(u_j^{(IIR)}[t]) \sum_{k=1}^{n_2} \delta_k^{(2)}[t] w_{k,j}^{(hidden)} \right) \cdot u_j^{(1)}[t - T + 1] \\ &= \sum_{j=1}^{n_1+1} \delta_j^{(IIR)}[t] \cdot u_j^{(2)}[t - T + 1] \quad (T = 1, \dots, N + 1) \end{aligned} \quad (28)$$

where $\delta_j^{(IIR)}[t] = g(u_j^{(IIR)}[t]) \sum_{k=1}^{n_2} \delta_k^{(2)}[t] w_{k,j}^{(hidden)}$.

For the quantum layer, the increment of the rotation angles θ is given by

$$\begin{aligned} \Delta \theta_{j,h} &= \sum_{T=1}^{N+1} \left(\frac{\partial e[t]}{\partial \alpha_j[t - T + 1]} \frac{\partial \alpha_j[t - T + 1]}{\partial \theta_{j,h}} \right. \\ &\quad \left. + \frac{\partial e[t]}{\partial \rho_j[t - T + 1]} \frac{\partial \rho_j[t - T + 1]}{\partial \theta_{j,h}} \right) \\ &\quad (j = 1, \dots, n_0, h = 1, \dots, n). \end{aligned} \quad (29)$$

We divide the calculation of $\Delta \theta_{j,h}$ into two steps. First, we calculate $\frac{\partial e[t]}{\partial \alpha_j[t_0]}$ and $\frac{\partial e[t]}{\partial \rho_j[t_0]}$, which are given by

$$\frac{\partial e[t]}{\partial \alpha_j[t_0]} = \frac{\partial e[t]}{\partial u_{2j-1}^{(IIR)}[t]} \frac{\partial u_{2j-1}^{(IIR)}[t]}{\partial u_{2j-1}^{(1)}[t_0]} \frac{\partial u_{2j-1}^{(1)}[t_0]}{\partial \alpha_j[t_0]}$$

$$\begin{aligned}
&= a_T \delta_{2j-1}^{(IRR)} \frac{\partial u_{2j-1}^{(1)}[t_0]}{\partial \alpha_j[t_0]} \\
&= -a_T \delta_{2j-1}^{(IRR)} \cdot 2 \sin\left(\frac{\pi}{2} f(\beta_j) - \alpha_j[t_0]\right) \\
&\quad \times \cos\left(\frac{\pi}{2} f(\beta_j) - \alpha_j[t_0]\right) \\
&= -a_T \delta_{2j-1}^{(IRR)} \cdot \sin(\pi f(\beta_j) - 2\alpha_j[t_0])
\end{aligned} \quad (30)$$

and

$$\begin{aligned}
\frac{\partial e[t]}{\partial \rho_j[t_0]} &= \frac{\partial e[t]}{\partial u_{2j}^{(IRR)}[t]} \frac{\partial u_{2j}^{(IRR)}[t]}{\partial u_{2j}^{(1)}[t_0]} \frac{\partial u_{2j}^{(1)}[t_0]}{\partial \rho_j[t_0]} \\
&= a_T \delta_{2j}^{(IRR)} \frac{\partial u_{2j}^{(1)}[t_0]}{\partial \rho_j[t_0]} \\
&= -a_T \delta_{2j}^{(IRR)} \cdot 2 \sin\left(\frac{\pi}{2} f(\sigma_j) - \rho_j[t_0]\right) \\
&\quad \times \cos\left(\frac{\pi}{2} f(\sigma_j) - \rho_j[t_0]\right) \\
&= -a_T \delta_{2j}^{(IRR)} \cdot \sin(\pi f(\sigma_j) - 2\rho_j[t_0]).
\end{aligned} \quad (31)$$

Next, we calculate $\frac{\partial \alpha_j[t_0]}{\partial \theta_{j,h}}$ by utilizing Eq. (14). First, we define some summation terms and partial derivative terms. The summation terms are given as follows:

$$a_j[t_0] = \operatorname{Re} \left(\sum_{h=1}^n \exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h}) \right) \quad (32)$$

$$b_j[t_0] = \operatorname{Im} \left(\sum_{h=1}^n \exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h}) \right) \quad (33)$$

$$c_j[t_0] = \operatorname{Re} \left(\sum_{h=1}^n \exp(i\phi_{j,h}) \cdot \cos(\varphi_h[t] + \theta_{j,h}) \right) \quad (34)$$

$$d_j[t_0] = \operatorname{Im} \left(\sum_{h=1}^n \exp(i\phi_{j,h}) \cdot \cos(\varphi_h[t] + \theta_{j,h}) \right). \quad (35)$$

The partial derivative terms are given as follows:

$$\frac{\partial a_j[t_0]}{\partial \theta_{j,h}} = \operatorname{Re} \left(\exp(i(\psi_h[t] + \phi_{j,h})) \cdot \cos(\varphi_h[t] + \theta_{j,h}) \right) \quad (36)$$

$$\frac{\partial b_j[t_0]}{\partial \theta_{j,h}} = \operatorname{Im} \left(\exp(i(\psi_h[t] + \phi_{j,h})) \cdot \cos(\varphi_h[t] + \theta_{j,h}) \right) \quad (37)$$

$$\frac{\partial c_j[t_0]}{\partial \theta_{j,h}} = \operatorname{Re} \left(-\exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h}) \right) \quad (38)$$

$$\frac{\partial d_j[t_0]}{\partial \theta_{j,h}} = \operatorname{Im} \left(-\exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h}) \right). \quad (39)$$

According to the division rule of complex numbers, $\frac{\partial \alpha_j[t_0]}{\partial \theta_{j,h}}$ is calculated by

$$\frac{\partial \alpha_j[t_0]}{\partial \theta_{j,i}} = \frac{1}{1 + \bar{I}_j^2[t_0]} \cdot \frac{\bar{P}_{j,i}[t_0] - \bar{Q}_{j,i}[t_0]}{\Omega_j^2[t_0]} \quad (40)$$

where

$$\Omega_j[t_0] = c_j^2[t_0] + d_j^2[t_0] \quad (41)$$

$$\bar{\Phi}_j[t_0] = a_j[t_0]c_j[t_0] + b_j[t_0]d_j[t_0] \quad (42)$$

$$\begin{aligned}
\bar{I}_j[t_0] &= \operatorname{Re} \left(\frac{\sum_{h=1}^n \exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h})}{\sum_{h=1}^n \exp(i\phi_{j,h}) \cdot \cos(\varphi_h[t] + \theta_{j,h})} \right) \\
&= \frac{\bar{\Phi}_j[t_0]}{\Omega_j[t_0]}
\end{aligned} \quad (43)$$

$$\begin{aligned}
\bar{P}_j[t_0] &= \Omega_j[t_0] \left(a_j[t_0] \frac{\partial c_j[t_0]}{\partial \theta_{j,h}} + c_j[t_0] \frac{\partial a_j[t_0]}{\partial \theta_{j,h}} \right. \\
&\quad \left. + b_j[t_0] \frac{\partial d_j[t_0]}{\partial \theta_{j,h}} + d_j[t_0] \frac{\partial b_j[t_0]}{\partial \theta_{j,h}} \right)
\end{aligned} \quad (44)$$

$$\bar{Q}_j[t_0] = 2\bar{\Phi}_j[t_0] \left(c_j[t_0] \frac{\partial c_j[t_0]}{\partial \theta_{j,h}} + d_j[t_0] \frac{\partial d_j[t_0]}{\partial \theta_{j,h}} \right). \quad (45)$$

At the same time, we calculate $\frac{\partial \rho_j[t_0]}{\partial \theta_{j,h}}$ by utilizing Eq. (15), which is given by

$$\frac{\partial \rho_j[t_0]}{\partial \theta_{j,i}} = \frac{1}{1 + \bar{I}_j^2[t_0]} \cdot \frac{\bar{P}_{j,i}[t_0] - \bar{Q}_{j,i}[t_0]}{\Omega_j^2[t_0]} \quad (46)$$

where

$$\Phi_j[t_0] = b_j[t_0]c_j[t_0] - a_j[t_0]d_j[t_0] \quad (47)$$

$$\begin{aligned}
\bar{I}_j[t_0] &= \operatorname{Im} \left(\frac{\sum_{h=1}^n \exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h})}{\sum_{h=1}^n \exp(i\phi_{j,h}) \cdot \cos(\varphi_h[t] + \theta_{j,h})} \right) \\
&= \frac{\Phi_j[t_0]}{\Omega_j[t_0]}
\end{aligned} \quad (48)$$

$$\begin{aligned}
P_j[t_0] &= \Omega_j[t_0] \left(c_j[t_0] \frac{\partial b_j[t_0]}{\partial \theta_{j,h}} + b_j[t_0] \frac{\partial c_j[t_0]}{\partial \theta_{j,h}} \right. \\
&\quad \left. - a_j[t_0] \frac{\partial d_j[t_0]}{\partial \theta_{j,h}} + d_j[t_0] \frac{\partial a_j[t_0]}{\partial \theta_{j,h}} \right)
\end{aligned} \quad (49)$$

$$\bar{Q}_j[t_0] = 2\Phi_j[t_0] \left(c_j[t_0] \frac{\partial c_j[t_0]}{\partial \theta_{j,h}} + d_j[t_0] \frac{\partial d_j[t_0]}{\partial \theta_{j,h}} \right). \quad (50)$$

The increment of the rotation angles \mathbf{f} is given by

$$\begin{aligned}
\Delta \phi_{j,h} &= \sum_{T=1}^{N+1} \left(\frac{\partial e[t]}{\partial \alpha_j[t-T+1]} \frac{\partial \alpha_j[t-T+1]}{\partial \phi_{j,h}} \right. \\
&\quad \left. + \frac{\partial e[t]}{\partial \rho_j[t-T+1]} \frac{\partial \rho_j[t-T+1]}{\partial \phi_{j,h}} \right) \\
&\quad (j = 1, \dots, n_0, h = 1, \dots, n).
\end{aligned} \quad (51)$$

Similar with $\Delta \theta_{j,h}$, we divide the calculation of $\Delta \phi_{j,h}$ into two steps. First, we calculate $\frac{\partial e[t]}{\partial \alpha_j[t_0]}$ and $\frac{\partial e[t]}{\partial \rho_j[t_0]}$, which are given in Eqs. (31) and (32). Next, we calculate $\frac{\partial \alpha_j[t_0]}{\partial \phi_{j,h}}$ and $\frac{\partial \rho_j[t_0]}{\partial \phi_{j,h}}$. Their calculations are similar to the calculations of $\frac{\partial \alpha_j[t_0]}{\partial \theta_{j,h}}$ and $\frac{\partial \rho_j[t_0]}{\partial \theta_{j,h}}$. What we need is to replace all $\theta_{j,h}$ by $\phi_{j,h}$; besides, the partial derivative terms given by Eqs. (37)–(40) should be substituted with the following:

$$\frac{\partial a_j[t_0]}{\partial \phi_{j,h}} = \operatorname{Re} \left(i \exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h}) \right) \quad (52)$$

$$\frac{\partial b_j[t_0]}{\partial \phi_{j,h}} = \operatorname{Im} \left(i \exp(i(\psi_h[t] + \phi_{j,h})) \cdot \sin(\varphi_h[t] + \theta_{j,h}) \right) \quad (53)$$

$$\frac{\partial c_j[t_0]}{\partial \phi_{j,h}} = \operatorname{Re} (i \exp (i (\psi_h[t] + \phi_{j,h})) \cdot \cos (\varphi_h[t] + \theta_{j,h})) \quad (54)$$

$$\frac{\partial d_j[t_0]}{\partial \phi_{j,h}} = \operatorname{Im} (i \exp (i (\psi_h[t] + \phi_{j,h})) \cdot \cos (\varphi_h[t] + \theta_{j,h})) \quad (55)$$

The increment of the quantum reversal vectors β and σ , is given by

$$\begin{aligned} \Delta \beta_j &= \frac{\partial e[t]}{\partial \beta_j} \\ &= \sum_{T=1}^{N+1} \left(\frac{\partial e[t]}{\partial u_{2j-1}^{(1)}[t-T+1]} \frac{\partial u_{2j-1}^{(1)}[t-T+1]}{\partial \beta_j} \right) \\ &= \sum_{T=1}^{N+1} \left(a_T \delta_{2j-1}^{(IRR)} \cdot 2 \sin \left(\frac{\pi}{2} f(\beta_j) - \alpha_j[t-T+1] \right) \right. \\ &\quad \times \cos \left(\frac{\pi}{2} f(\beta_j) - \alpha_j[t-T+1] \right) \cdot \frac{\pi}{2} g(\beta_j) \left. \right) \\ &= \frac{\pi}{2} a_T \delta_{2j-1}^{(IRR)} g(\beta_j) \cdot \sum_{T=1}^{N+1} \sin (\pi f(\beta_j) - 2\alpha_j[t-T+1]) \quad (56) \end{aligned}$$

$$\begin{aligned} \Delta \sigma_j &= \frac{\partial e[t]}{\partial \sigma_j} \\ &= \sum_{T=1}^{N+1} \left(\frac{\partial e[t]}{\partial u_{2j}^{(1)}[t-T+1]} \frac{\partial u_{2j}^{(1)}[t-T+1]}{\partial \sigma_j} \right) \\ &= \sum_{T=1}^{N+1} \left(a_T \delta_{2j}^{(IRR)} \cdot 2 \sin \left(\frac{\pi}{2} f(\sigma_j) - \rho_j[t-T+1] \right) \right. \\ &\quad \times \cos \left(\frac{\pi}{2} f(\sigma_j) - \rho_j[t-T+1] \right) \cdot \frac{\pi}{2} g(\sigma_j) \left. \right) \\ &= \frac{\pi}{2} a_T \delta_{2j}^{(IRR)} g(\sigma_j) \cdot \sum_{T=1}^{N+1} \sin (\pi f(\sigma_j) - 2\rho_j[t-T+1]) \quad (57) \end{aligned}$$

References

- Abarbanel, H. (1996). *Analysis of observed chaotic data*. Springer.
- Ardalani-Farsa, M., & Zolfaghari, S. (2010). Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks. *Neurocomputing*, 73, 2540–2553.
- Ayoubi, M. (1994). Fault diagnosis with dynamic neural structure and application to a turbo-charger. In *International symposium on fault detection, supervision and safety for technical processes* (pp. 13–16). Espoo, Finland.
- Back, A. D., & Tsoi, A. C. (1991). FIR and IIR synapses, a new neural network architecture for time series modeling. *Neural Computation*, 3, 375–385.
- Benioff, P. (1982). Quantum mechanical Hamiltonian models of Turing machines. *Journal of Statistical Physics*, 29, 515–546.
- Chandra, R., & Zhang, M. (2012). Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction. *Neurocomputing*, 86, 116–123.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14, 179–211.
- Ezhov, A. A., & Ventura, D. (2000). Quantum neural networks. In *Future directions for intelligent systems and information sciences* (pp. 213–235). Springer.
- Feynman, R. P. (1982). Simulating physics with computers. *International journal of theoretical physics*, 21, 467–488.
- Frasconi, P., Gori, M., & Soda, G. (1992). Local feedback multilayered networks. *Neural Computation*, 4, 120–130.
- Gholipour, A., Araabi, B. N., & Lucas, C. (2006). Predicting chaotic time series using neural and neurofuzzy models: a comparative study. *Neural Processing Letters*, 24, 217–239.
- Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18, 602–610.
- Grover, L. K. (1997). Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79, 325.
- Gu, J., Lau, D., & Pecht, M. (2009). Health assessment and prognostics of electronic products. In: *Proceedings of 8th international conference on reliability, maintainability and safety* (pp. 21–25).
- Han, K.-H., & Kim, J.-H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *Evolutionary Computation, IEEE Transactions on*, 6, 580–593.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79, 2554–2558.

- Jozsa, R., & Linden, N. (2003). On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 459, 2011–2032.
- Kabir, A., Bailey, C., Lu, H., & Stoyanov, S. (2012). A review of data-driven prognostics in power electronics. In *Electronics technology (ISSE), 2012 35th international spring seminar on* (pp. 189–192). IEEE.
- Kak, S. C. (1995). Quantum neural computing. *Advances in Imaging and Electron Physics*, 94, 259–313.
- Kasabov, N. (2010). Integrative probabilistic evolving spiking neural networks utilising quantum inspired evolutionary algorithm: a computational framework. In J. Koronacki, Z. Raś, S. Wierzbicki, & J. Kacprzyk (Eds.), *Advances in Machine Learning II: Vol. 263* (pp. 415–425). Berlin Heidelberg: Springer.
- Kasabov, N., Dhole, K., Nuntalid, N., & Indiveri, G. (2013). Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. *Neural Networks*, 41, 188–201.
- Kasabov, N.K. (2007). *Evolving connectionist systems: the knowledge engineering approach*. Springer Science & Business Media.
- Kim, J. H. (1997). Time-varying two-phase optimization neural network. *Journal of Intelligent and Fuzzy Systems*, 5, 85–101.
- Koskela, T., Lehtokangas, M., Saarinen, J., & Kaski, K. (1996). Time series prediction with multilayer perceptron, FIR and Elman neural networks. In: *Proceedings of the world congress on neural networks* (pp. 491–496).
- Kouda, N., Matsui, N., & Nishimura, H. (2002). Image compression by layered quantum neural networks. *Neural Processing Letters*, 16, 67–80.
- Kouda, N., Matsui, N., Nishimura, H., & Peper, F. (2005). Qubit neural network and its learning efficiency. *Neural Computing & Applications*, 14, 114–121.
- Kuk-Hyun, H., & Jong-Hwan, K. (2004). Quantum-inspired evolutionary algorithms with a new termination criterion, H-sub>εpsi;</sub>-gate, and two-phase scheme. *Evolutionary Computation, IEEE Transactions on*, 8, 156–169.
- Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1997). Artificial neural network methods in quantum mechanics. *Computer Physics Communications*, 104, 1–14.
- Lebedev, D. V., Steil, J. J., & Ritter, H. J. (2005). The dynamic wave expansion neural network model for robot motion planning in time-varying environments. *Neural Networks*, 18, 267–285.
- Li, P., Li, Y., Xiong, Q., Chai, Y., & Zhang, Y. (2014). Application of a hybrid quantized Elman neural network in short-term load forecasting. *International Journal of Electrical Power & Energy Systems*, 55, 749–759.
- Li, P., & Xiao, H. (2014). Model and algorithm of quantum-inspired neural network with sequence input based on controlled rotation gates. *Applied Intelligence*, 40, 107–126.
- Li, P., Xiao, H., Shang, F., Tong, X., Li, X., & Cao, M. (2013). A hybrid quantum-inspired neural networks with sequence inputs. *Neurocomputing*, 117, 81–90.
- Liu, C.-Y., Chen, C., Chang, C.-T., & Shih, L.-M. (2013). Single-hidden-layer feed-forward quantum neural network based on Grover learning. *Neural Networks*, 45, 144–150.
- Lou, X., & Cui, B. (2007). Synchronization of competitive neural networks with different time scales. *Physica A: Statistical Mechanics and its Applications*, 380, 563–576.
- Mandic, D., & Chambers, J. (2001). *Recurrent neural networks for prediction: Architectures, learning algorithms and stability*. New York: Wiley.
- Manju, A., & Nigam, M. J. (2014). Applications of quantum inspired computational intelligence: a survey. *Artificial Intelligence Review*, 42, 79–156.
- Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11, 431–441.
- Marra, S., & Morabito, F. C. (2006). Solar activity forecasting by incorporating prior knowledge from nonlinear dynamics into neural networks. In *Neural networks, 2006. international joint conference on* (pp. 3722–3728). IEEE.
- Matsui, N., Nishimura, H., & Peper, F. (2005). An examination of qubit neural network in controlling an inverted pendulum. *Neural Processing Letters*, 22, 277–290.
- Matsui, N., Takai, M., & Nishimura, H. (2000). A network model based on qubitlike neuron corresponding to quantum circuit. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 83, 67–73.
- Menneer, T., & Narayanan, A. (1995). *Quantum-inspired neural networks*. UK: Department of Computer Sciences, University of Exeter.
- Menneer, T. S. I. (1999). *Quantum artificial neural networks*. (Ph. D thesis). University of Exeter.
- Meyer-Bäse, A., Ohl, F., & Scheich, H. (1996). Singular perturbation analysis of competitive neural networks with different time scales. *Neural Computation*, 8, 1731–1742.
- Pecht, M., & Jaai, R. (2010). A prognostics and health management roadmap for information and electronics-rich systems. *Microelectronics Reliability*, 50, 317–323.
- Platel, M. D., Schliebs, S., & Kasabov, N. (2009). Quantum-inspired evolutionary algorithm: a multimodel EDA. *Evolutionary Computation, IEEE Transactions on*, 13, 1218–1232.
- Purushothaman, G., & Karayiannis, N. B. (1997). Quantum neural networks (QNNs): inherently fuzzy feedforward neural networks. *Neural Networks, IEEE Transactions on*, 8, 679–693.
- Rigatos, G. G., & Tzafestas, S. G. (2002). Parallelization of a fuzzy control algorithm using quantum computation. *Fuzzy Systems, IEEE Transactions on*, 10, 451–460.
- Sarathi Vasan, A. S., Long, B., & Pecht, M. (2013). Diagnostics and prognostics method for analog electronic circuits. *Industrial Electronics, IEEE Transactions on*, 60, 5277–5291.
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45, 2673–2681.

- Seising, R. (2006). Can Fuzzy Sets Be Useful in the (Re) Interpretation of Uncertainty in Quantum Mechanics?. In *Fuzzy information processing society, 2006. Annual meeting of the North American* (pp. 414–419). IEEE.
- Sello, S. (2001). Solar cycle forecasting: a nonlinear dynamics approach. *Astronomy and Astrophysics-Berlin-*, 377, 312–320.
- Shafee, F. (2007). Neural networks with quantum gated nodes. *Engineering Applications of Artificial Intelligence*, 20, 429–437.
- Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26, 1484–1509.
- SIDC. (2015). *World Data Center for the Sunspot Index, Monthly Smoothed Sunspot Data*. <http://sidc.oma.be>.
- Svitek, M. (2008). Wave probabilities and quantum entanglement. *Neural Network World*, 18, 401–406.
- Teo, K. K., Wang, L., & Lin, Z. (2001). Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization. In *Computational science-ICCS 2001* (pp. 310–317). Springer.
- Widiputra, H., Pears, R., & Kasabov, N. (2011). Multiple time-series prediction through multiple time-series relationships profiling and clustered recurring trends. In *Advances in knowledge discovery and data mining* (pp. 161–172). Springer.
- Yazdizadeh, A., & Khorasani, K. (2002). Adaptive time delay neural network structures for nonlinear system identification. *Neurocomputing*, 47, 207–240.
- Zak, M., & Williams, C. P. (1998). Quantum neural nets. *International Journal of Theoretical Physics*, 37, 651–684.
- Zhang, G., Eddy Patuwo, B., & Y Hu, M. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14, 35–62.