CHAPTER

ECHO STATE NETWORKS FOR MULTIDIMENSIONAL DATA: EXPLOITING NONCIRCULARITY AND WIDELY LINEAR MODELS

12

Yili Xia[*], **Min Xiang**[†], **Zhe Li**[*], **Danilo P. Mandic**[†]

[*]*Southeast University, School of Information Science and Engineering, Nanjing, P. R. China* [†]*Imperial College London, London, United Kingdom*

CONTENTS

CHAPTER POINTS

- Quaternion-valued echo state networks (QESNs) are introduced for nonlinear adaptive filtering of 3D and 4D signals.
- The exploitation of full quaternion nonlinear activation functions with local analyticity allows for first-order differentiability at the operating point to design gradient descent–based quaternion nonlinear learning algorithms.
- The widely linear model is incorporated into QESNs for optimal performance for second-order noncircular quaternion signals.

Adaptive Learning Methods for Nonlinear System Modeling. DOI: 10.1016/B978-0-12-812976-0.00016-6

12.1 INTRODUCTION

Recurrent neural networks (RNNs) represent a large and varied class of computational models that are designed by more or less detailed analogy with biological brain modules. In RNNs, numerous abstract neurons are interconnected by abstracted synaptic connections, which enable activations to propagate through the network. The characteristic feature of RNNs that distinguishes them from the feedforward neural networks is that the connection topology possesses feedback cycles. The computational power of RNNs stems from their ability to act as universal approximators for any continuous function on a compact domain [1,2], and owing to their rich inherent memory through feedback, RNNs have found wide applications in the modeling of highly nonlinear and nonstationary dynamic systems and the associated attractor dynamics, including system identification [3], time series prediction [4] and adaptive noise cancellation [5], where, for the nonstationary and nonlinear nature of the signals and typically long impulse responses, using the class of static feedforward networks or transversal filters would result in undermodeling [2,6].

Recent progress in sensing technology has made possible the recording from multidimensional data sources which are typically 3D and 4D, such as measurements from inertial sensors in body sensor networks, three-axis ultrasonic anemometers in wind measurement and three-axis seismometers in oil exploration. Real-valued multichannel signal processing considers these measurements as vectors in $\mathbb{R}^3$ and $\mathbb{R}^4$ and cater for the “coupled” nature of the available information across data channels, but the information in the correlation matrices is not easy to use, due to their multiple and scattered block structures. In practice, when 4D data are modeled using real-valued vectors, they produce e.g., 10 covariance matrices. Also, accuracy may be compromised due to the deficiencies of the nondivision vector algebra (gimbal lock). These problems are largely mitigated when modeling 3D and 4D signals by quaternions, a number system first introduced by William Rowan Hamilton in 1843, since its 4D division algebra naturally accounts for the coupling between data channels and also provides parsimonious representations, e.g., with only four covariance matrices needed. For this reason, quaternions, which were traditionally used in aerospace engineering, because of their advantage in modeling 3D rotations and orientations, are rapidly gaining popularity in various applications, including computer graphics [7], molecular modeling [8], color image processing [9,10] and 3D polarized signal representation for vector sensor array processing [11,12].

In the context of learning systems, quaternion approaches include both Kalman filtering [13–15] and stochastic gradient algorithms, such as the quaternion least mean square [16]. However, in the context of nonlinear learning systems like RNNs, quaternion-valued processing is still relatively underexplored, mainly because of problems arising from the lack of analytic nonlinearities in the quaternion domain $\mathbb{H}$. Although quaternion nonlinear functions have been implemented, for example, in the quaternion independent component analysis [17], the analyticity of these functions has not been rigorously examined, mainly because the very stringent Cauchy–Riemann–Fueter (CRF) conditions admit only linear functions and constants as globally analytic quaternion-valued functions [18]. This is a serious obstacle that prevents us from choosing the standard nonlinear activation functions (tanh, logistic) as the nonlinearities in the design of quaternion-valued nonlinear learning algorithms.

To partially overcome this issue, some approaches use a “split” quaternion function that treats each quaternion component separately (as a real channel), passed through a real-valued smooth nonlinearity [19,20]. Although this may yield enhanced performance over vector-based processing, the noncommutativity aspect of the quaternion algebra is overlooked, thus prohibiting rigorous treatment of the cross-information and not exploiting the full potential of quaternions. Recognizing that gradient-based learning algorithms, such as nonlinear gradient descent (NGD) and real-time recurrent learning (RTRL) [2], require gradient evaluation at a point, it is possible to adopt a local alternative, such as the local analyticity condition (LAC) [21], to relax the global analyticity requirement for quaternion nonlinearities by the CRF conditions. The work in [22] uses LAC to establish a class of neural networks in $\mathbb{H}$, which are a generic extension of those in $\mathbb{R}$ and $\mathbb{C}$, allowing us to use standard activation functions, such as tanh. The learning algorithms introduced in this way include quaternion nonlinear gradient decent for nonlinear FIR filters [22] and quaternion real-time recurrent learning for fully connected neural networks [23]. However, another obvious obstacle which hinders the further development of quaternion RNNs is the high computational complexity associated with their training, as a quaternion addition requires four real-valued additions whereas a quaternion multiplication requires 16 real-valued multiplications and 12 real-valued additions.

Recently, a class of discrete-time RNNs, called echo state networks (ESNs), has been introduced, with the aim to reduce the complexity of computation encountered by standard RNNs [24,25]. The principle behind ESNs is to separate the RNN architecture into two constituent components: a recurrent architecture, called the "dynamical reservoir" or "hidden layer" and a memoryless output layer, called the "readout neuron." The recurrent architecture consists of a randomly generated group of hidden neurons with a specified degree of recurrent connections and should satisfy the so-called "echo state property" to maintain stability [24,26]. This way, the high computational complexity of RNNs is significantly reduced due to the sparse connections among the hidden neurons; in addition, the learning requirements are reduced to only the weights connecting the hidden layer and the readout neuron, and the emergence of ESNs enables the possibility to design computationally efficient quaternion-valued RNNs with moderate complexity.

As quaternions can be conveniently constructed as a pair of complex numbers through the Cayley–Dickson representation, it is natural to ask whether some recent breakthroughs in statistical signal processing of complex variables can be generalized to quaternions. One such breakthrough in complex-valued signal processing has been due to the widely linear model and augmented complex statistics [27,28]. This model, together with the corresponding augmented complex statistics, has been successfully used to design enhanced algorithms in communications and adaptive filters [29,30]. A number of studies have shown that widely linear modeling offers theoretical and practical advantages over the standard strictly linear model, and it is applicable to the generality of complex-valued random signals, both second-order circular (proper) and noncircular (improper) [28,31–33]. The concept of $\mathbb{H}$ properness was introduced in [34] as the invariance of the probability density function (pdf) of a quaternion-valued variable q under some specific rotations around the angle of $\pi/2$, and this restriction was later relaxed to an arbitrary axis and angle of rotation ϕ [35]. In this sense, a variable q is said to be proper, if $\text{pdf}(q) = \text{pdf}(e^{p\phi}q)$ for any pure unit quaternion p. These discoveries were followed by the quaternion widely linear model, together with the augmented quaternion statistics, to account for the generality of quaternion signals, both proper and improper [36–39]. Such models have been shown to outperform the traditional strictly linear quaternion processing, which relies only on the standard covariance matrix, and a rigorous quantitative analysis of statistical advantages of widely linear over strictly linear processing in $\mathbb{H}$ has been performed in [40].

Therefore, in this chapter, we introduce quaternion-valued echo state networks (QESNs), as a generic extension of real-valued and complex-valued ESNs [24,25,41], to enable the processing of hypercomplex 3D and 4D signals. To further equip the QESNs with enhanced modeling capability for noncircular quaternion signals (rotation-dependent distribution), we incorporate the widely linear model into QESNs, making them second-order optimal for the generality of quaternion signals (both circular and noncircular) [42]. The so introduced widely linear model for state space estimation is not limited to QESNs but is also applicable to general quaternion-valued RNN architectures. Simulations and experimental results in the prediction setting on both benchmark circular and noncircular signals and on real-world noncircular, nonlinear and nonstationary 3D body motion and wind data support the analysis.

12.2 MATHEMATICAL BACKGROUND

The necessary mathematical background on the design of QESNs and the associated nonlinear learning algorithms will be briefly discussed in this section.

12.2.1 QUATERNION ALGEBRA

As a noncommutative extension of the complex domain $\mathbb{C}$, the quaternion domain $\mathbb{H}$ provides a natural framework for processing 3D and 4D signals. A quaternion variable $q \in \mathbb{H}$ is a skew field over $\mathbb{R}$ and comprises four real-valued components (q_r, $q_\imath$, $q_\jmath$ and q_κ) and three imaginary units ($\imath$, $\jmath$ and κ), to give

$$q = q_r + \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa = Sq + Vq. \tag{12.1}$$

It can also be represented by a real part (scalar), denoted by $Sq = \Re(q) = q_r$, and a vector part Vq (also known as a pure quaternion $\Im(q)$), consisting of the three imaginary components, $Vq = \Im(q) = \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa$. The pure quaternions $\Im(q)$ are usually used for 3D signal modeling. The imaginary units $\imath$, $\jmath$ and κ obey the following rules:

$$\imath\jmath = \kappa, \jmath\kappa = i, \kappa\imath = \jmath, \imath^2 = \jmath^2 = \kappa^2 = \imath\jmath\kappa = -1. \tag{12.2}$$

Note that the quaternion multiplication is not commutative, that is, $\imath\jmath \neq \jmath\imath = -\kappa$. For any two quaternions, q_1 and q_2, their product is given by

$$\begin{aligned} q_1 q_2 &= (Sq_1 + Vq_1)(Sq_2 + Vq_2) \\ &= Sq_1 Sq_2 - Vq_1 \cdot Vq_2 + Sq_1 Vq_2 + Sq_2 Vq_1 + Vq_1 \times Vq_2, \end{aligned} \tag{12.3}$$

where the symbol "·" denotes the dot-product and "×" the cross-product. The cross-product makes the quaternion multiplication noncommutative. The norm of a quaternion variable q is given by

$$\|q\| = \sqrt{qq^*} = \sqrt{q_r^2 + q_\imath^2 + q_\jmath^2 + q_\kappa^2} \tag{12.4}$$

while the quaternion conjugate of q, denoted by q^*, is given by

$$q^* = Sq - Vq = q_r - \imath q_\imath - \jmath q_\jmath - \kappa q_\kappa. \tag{12.5}$$

Another important notion in the quaternion domain is the so-called "quaternion involution," which defines a self-inverse mapping, analogous to the complex conjugate. The general involution of a quaternion variable q is defined as $\phi_\alpha(q) \equiv q^\alpha = \alpha q \alpha^{-1} = -\alpha q \alpha$, where $\alpha \in \mathbb{H}$, while the special cases of involutions about the $\imath$, $\jmath$ and κ imaginary axes are given by [9]

$$\begin{aligned} q^\imath &= -\imath q \imath = q_r + \imath q_\imath - \jmath q_\jmath - \kappa q_\kappa, \\ q^\jmath &= -\jmath q \jmath = q_r - \imath q_\imath + \jmath q_\jmath - \kappa q_\kappa, \\ q^\kappa &= -\kappa q \kappa = q_r - \imath q_\imath - \jmath q_\jmath + \kappa q_\kappa, \end{aligned} \tag{12.6}$$

with the following properties ($\forall \eta \in \{\imath, \jmath, \kappa\}$):

$$(q^\eta)^\eta \;=\; q,\ (q^\eta)^* = (q^*)^\eta,\ (q_1 q_2)^\eta = q_1^\eta q_2^\eta,\ (q_1{+}q_2)^\eta = q_1^\eta{+}q_2^\eta. \tag{12.7}$$

Similar to the complex conjugate, involutions allow for the components of a quaternion variable q to be expressed in terms of the actual variable q and its "partial conjugate," $q^\imath$, $q^\jmath$ and q^κ, that is,

$$\begin{aligned} q_r &= \frac{1}{4}(q + q^\imath + q^\jmath + q^\kappa),\ q_\imath = \frac{1}{4\imath}(q + q^\imath - q^\jmath - q^\kappa), \\ q_\jmath &= \frac{1}{4\jmath}(q - q^\imath + q^\jmath - q^\kappa),\ q_\kappa = \frac{1}{4\kappa}(q - q^\imath - q^\jmath + q^\kappa). \end{aligned} \tag{12.8}$$

In this way, the relationship between the involutions and the quaternion variable q and its conjugate is specified by

$$q = \frac{1}{2}(q^{\imath *} + q^{\jmath *} + q^{\kappa *} - q^*),\ q^* = \frac{1}{2}(q^\imath + q^\jmath + q^\kappa - q). \tag{12.9}$$

12.2.2 QUATERNION WIDELY LINEAR MODEL AND QUATERNION AUGMENTED STATISTICS

Now, consider a quaternion-valued mean square error (MSE) estimator given by

$$\hat{y} = E[y|\mathbf{q}], \tag{12.10}$$

where $\hat{y}$ is the estimated process, $\mathbf{q}$ is the observed variable and $E[\cdot]$ is the statistical expectation operator. For zero mean jointly Gaussian $\mathbf{q}$ and y, the strictly linear estimation solution, similar to those in $\mathbb{R}$ and $\mathbb{C}$, is given by [13,16]

$$\hat{y} = \mathbf{w}^{\mathrm{T}}\mathbf{q}, \tag{12.11}$$

where $\mathbf{w}$ and $\mathbf{q}$ are, respectively, the coefficient and regressor vector. Observe, however, that for all the components $\{y_r, y_\imath, y_\jmath, y_\kappa\}$, we have

$$\hat{y}_\eta = E[y_\eta|\mathbf{q}_r, \mathbf{q}_\imath, \mathbf{q}_\jmath, \mathbf{q}_\kappa],\quad \eta \in \{r, \imath, \jmath, \kappa\}, \tag{12.12}$$

so that by using the involutions in (12.6), we can express each element of a quaternion variable as in (12.8). This gives, for instance, the real component of a quaternion variable $\mathbf{q}_r = (\mathbf{q} + \mathbf{q}^\imath + \mathbf{q}^\jmath + \mathbf{q}^\kappa)/4$, leading to the general expression for all the components

$$\hat{y}_\eta = E[y_\eta|\mathbf{q}, \mathbf{q}^\imath, \mathbf{q}^\jmath, \mathbf{q}^\kappa] \text{ and } \hat{y} = E[y|\mathbf{q}, \mathbf{q}^\imath, \mathbf{q}^\jmath, \mathbf{q}^\kappa] \tag{12.13}$$

and allowing us to arrive at the widely linear model [36–38]

$$y = \mathbf{w}^{a\mathrm{T}}\mathbf{q}^a = \mathbf{a}^\mathrm{T}\mathbf{q} + \mathbf{b}^\mathrm{T}\mathbf{q}^\imath + \mathbf{c}^\mathrm{T}\mathbf{q}^\jmath + \mathbf{d}^\mathrm{T}\mathbf{q}^\kappa, \tag{12.14}$$

where $\mathbf{w}^a = [\mathbf{a}^\mathrm{T}, \mathbf{b}^\mathrm{T}, \mathbf{c}^\mathrm{T}, \mathbf{d}^\mathrm{T}]^\mathrm{T}$ is the augmented weight vector.

Current statistical signal processing in $\mathbb{H}$ by and large employs the strictly linear model in (12.11), drawing upon the covariance matrix $\mathbf{R} = E[\mathbf{q}\mathbf{q}^\mathrm{H}]$, which is sufficient to describe only second-order circular (proper) signals with equal power data components. However, based on (12.14), the modeling of both the second-order circular (proper) and noncircular (improper) signals is only possible using the augmented covariance matrix, given by [36–39],

$$\mathbf{R}^a = E[\mathbf{q}^a\mathbf{q}^{a\mathrm{H}}] = \begin{bmatrix} \mathbf{R} & \mathbf{P} & \mathbf{S} & \mathbf{T} \\ \mathbf{P}^\imath & \mathbf{R}^\imath & \mathbf{T}^\imath & \mathbf{S}^\imath \\ \mathbf{S}^\jmath & \mathbf{T}^\jmath & \mathbf{R}^\jmath & \mathbf{P}^\jmath \\ \mathbf{T}^\kappa & \mathbf{S}^\kappa & \mathbf{P}^\kappa & \mathbf{R}^\kappa \end{bmatrix}, \tag{12.15}$$

where $\mathbf{R} = E[\mathbf{q}\mathbf{q}^H]$ and the three pseudocovariances are termed the $\imath$-covariance $\mathbf{P}$, $\jmath$-covariance $\mathbf{S}$ and κ-covariance $\mathbf{T}$, given by

$$\mathbf{P} = E[\mathbf{q}\mathbf{q}^{\imath\mathrm{H}}], \quad \mathbf{S} = E[\mathbf{q}\mathbf{q}^{\jmath\mathrm{H}}], \quad \mathbf{T} = E[\mathbf{q}\mathbf{q}^{\kappa\mathrm{T}}]. \tag{12.16}$$

Second-order circular (proper) quaternion-valued signals have probability distributions that are rotation-invariant with respect to all the six possible pairs of axes (combinations of $\imath$, $\jmath$ and κ) [38] and thus equal powers in all the components, so that the three pseudocovariance matrices $\mathbf{P}$, $\mathbf{S}$ and $\mathbf{T}$ vanish. However, in most real-world applications, probability density functions are rotation-dependent and hence require the use of the augmented quaternion statistics and the quaternion widely linear model. Also note that real-valued multichannel processing in $\mathbb{R}^4$ requires ten covariance matrices, as opposed to four in the quaternion domain, since only $\mathbf{R}$, $\mathbf{P}$, $\mathbf{S}$ and $\mathbf{T}$ are needed to fully describe $\mathbf{R}^a$ in (12.15).

12.2.3 $\mathbb{HR}$-CALCULUS AND QUATERNION GRADIENT OPERATIONS

In the design of quaternion-valued adaptive learning systems, a common optimization objective is to minimize a positive real function of quaternion variables. However, gradient-based optimization in $\mathbb{H}$ is quite restrictive, as the standard CRF conditions do not admit derivatives of nonanalytic real-valued cost functions [18]. It is only recently that the $\mathbb{HR}$-calculus has made possible the differentiation of both analytic and nonanalytic functions in $\mathbb{H}$ [43], through exploiting the duality with their isomorphic quadrivariate real functions, which is built upon the relationships between quaternion involutions and their four real-valued components, as indicated by (12.6). Compared with other existing quaternion gradients, which typically consider the real part and the vector part of a quaternion separately [21, 44], the $\mathbb{HR}$-calculus enables the possibility to treat a quaternion as an entry, giving more physically intuitive gradient calculation results with compact expressions.

The $\mathbb{HR}$- and $\mathbb{HR}^*$-derivatives are respectively given by [43]

$$\frac{\partial f(q,q^{\imath},q^{\jmath},q^{\kappa})}{\partial q}=\frac{1}{4}\left(\frac{\partial f}{\partial q_r}-\imath\frac{\partial f}{\partial q_\imath}-\jmath\frac{\partial f}{\partial q_\jmath}-\kappa\frac{\partial f}{\partial q_\kappa}\right) \tag{12.17}$$

and

$$\frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^*}=\frac{1}{4}\left(\frac{\partial f}{\partial q_r}+\imath\frac{\partial f}{\partial q_\imath}+\jmath\frac{\partial f}{\partial q_\jmath}+\kappa\frac{\partial f}{\partial q_\kappa}\right). \tag{12.18}$$

We shall now consider some examples on how to use the $\mathbb{HR}$-calculus for derivatives typically encountered in the design of quaternion-valued RNNs.

- *Derivative of the analytic function* $f(q)=q$. We first express it in terms of the involutions $\{q,q^{\imath},q^{\jmath},q^{\kappa}\}$ using (12.9) and then differentiate it with respect to q, giving $\partial f(q)/\partial q=1$. This is equivalent to the standard CRF derivative, which gives $f'(q)=1$.
- *Derivative of the nonanalytic function* $f(q,q^*)=qq^*$. Such a cost function is dependent on both q and its conjugate q^* and is nonanalytic in the CRF sense. However, the $\mathbb{HR}$-calculus circumvents this problem through (12.17) and (12.18). For example, consider $\partial f(q,q^*)/\partial q^*=\partial(qq^*)/\partial q^*=\partial q/\partial q^*\cdot q^*+q\cdot\partial q^*/\partial q^*$. The $\mathbb{HR}^*$-derivative gives $\partial q/\partial q^*=-1/2$ and the $\mathbb{HR}$-derivative $\partial q^*/\partial q^*=\partial q/\partial q=1$, and hence $\partial f(q,q^*)/\partial q^*=\partial(qq^*)/\partial q^*=\partial q/\partial q^*\cdot q^*+q\cdot\partial q^*/\partial q^*=-q^*/2+q$.
- *Derivatives of involutions* $f(q)=q^{\eta}$, $\eta=\{\imath,\jmath,\kappa\}$. As desired, the $\mathbb{HR}$-derivatives give

$$\frac{\partial q^{\eta}}{\partial q^{\theta}}=\begin{cases}1, & \eta=\theta,\quad \eta,\theta\in\{\imath,\jmath,\kappa\},\\ 0, & \eta\neq\theta,\quad \eta,\theta\in\{\imath,\jmath,\kappa\}.\end{cases} \tag{12.19}$$

12.2.4 QUATERNION NONLINEAR ACTIVATION FUNCTIONS

Hypercomplex nonlinear learning systems, like quaternion-valued RNNs, are more difficult to design than their real-valued counterparts, owing to the lack of analytic nonlinear activation functions. This is due to the stringent CRF conditions [18]. For instance, a CRF differentiable quaternion function $f(q)$ should satisfy

$$\frac{\partial f}{\partial q_r}+\imath\frac{\partial f}{\partial q_\imath}+\jmath\frac{\partial f}{\partial q_\jmath}+\kappa\frac{\partial f}{\partial q_\kappa}=0\Leftrightarrow\frac{\partial f}{\partial q^*}=0. \tag{12.20}$$

It is important to note that only linear quaternion functions and constants fulfill these conditions, yet nonlinear learning algorithms in $\mathbb{H}$ require differentiable nonlinear functions. To circumvent the analyticity problem, recent work in [22] adopted the local analyticity condition (LAC) [21], based on a complex-valued representation of a quaternion, to give

$$\frac{\partial f}{\partial q_r}=-\zeta\frac{\partial f}{\partial \alpha}, \tag{12.21}$$

where ζ and α are respectively given by

$$\zeta = \frac{\imath q_\imath + \jmath q_\jmath + \kappa q_\kappa}{\alpha}, \quad \alpha = \sqrt{q_\imath^2 + q_\jmath^2 + q_\kappa^2}. \tag{12.22}$$

In this way, an "imaginary" unit ζ comprises the vector part of quaternions. Although the local analyticity condition only guarantees first-order differentiability at the current operating point, this is a perfect match for gradient descent–based quaternion-valued learning algorithms, which require gradient evaluation at a point only.

Proposition 1. *The quaternion exponential* $e^q = e^{q_r + \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa}$ *satisfies the LAC in* (12.21).

Proof. We can expand e^q using the Euler formula as

$$\begin{aligned} e^q &= e^{q_r}\left(\cos(\alpha) + \zeta \sin(\alpha)\right) \\ &= e^{q_r}\left(\cos(\alpha) + \frac{\imath q_\imath \sin(\alpha)}{\alpha} + \frac{\jmath q_\jmath \sin(\alpha)}{\alpha} + \frac{\kappa q_\kappa \sin(\alpha)}{\alpha}\right), \end{aligned} \tag{12.23}$$

where ζ and α are defined in (12.22), to give

$$\frac{\partial e^q}{\partial q_r} = e^q = -\zeta \frac{e^q}{\alpha}. \tag{12.24}$$

Note that the quaternion exponential $e^{-q} = e^{-(q_r + \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa)}$ also satisfies the LAC in (12.21). This is straightforward to show using the same approach as in **Proposition 1**. In this way, quaternion transcendental nonlinear functions, constructed on the basis of quaternion exponentials e^q and e^{-q}, are a generic extension of those in $\mathbb{R}$ and $\mathbb{C}$, and they also satisfy the LAC [22]. In this chapter, we employ a fully quaternion hyperbolic tangent function to design the quaternion ESNs, defined as [22]

$$\tanh(q) = \frac{e^q - e^{-q}}{e^q + e^{-q}} = \frac{e^{2q} - 1}{e^{2q} + 1}, \tag{12.25}$$

for which the first derivative is given by

$$\frac{\partial \tanh(q)}{\partial q} = \text{sech}^2(q), \quad \text{sech}(q) = \frac{2}{e^q + e^{-q}}. \tag{12.26}$$

12.3 QUATERNION ESNS

The existence of fully quaternion nonlinear activation functions enables the design of RNNs in $\mathbb{H}$ [22, 23]. In this section, the quaternion ESNs (QESNs) and its augmented (widely linear) variant (AQESNs) are introduced.

12.3.1 STANDARD QESNS

The architecture of a standard ESN, which is a recurrent discrete-time neural network composed of K external inputs, N internal neurons (also referred to as dynamical reservoir) and L readout neurons, is

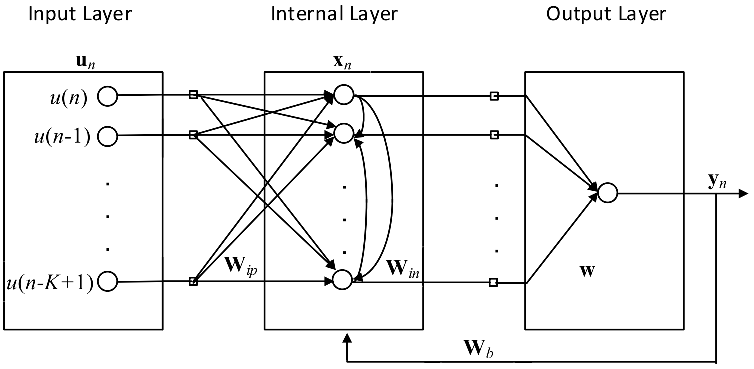

FIGURE 12.1

The architecture of an Echo State Network (ESN).

shown in Fig. 12.1. The $N \times K$ weight matrix $\mathbf{W}_{ip}$ and the $N \times N$ weight matrix $\mathbf{W}_{in}$ respectively contain the connections between the input units and the internal units and those between the internal units themselves, whereas the feedback connections between the internal neurons and the readout neurons are stored in the $N \times L$ feedback weight matrix $\mathbf{W}_b$. The $N \times 1$ internal state vector $\mathbf{x}_n$, $K \times 1$ input vector $\mathbf{u}_n$ and $L \times 1$ output vector $\mathbf{y}_n$ are all at time instant n. By concatenating the input $\mathbf{u}_n$, the internal state $\mathbf{x}_n$ and the delayed output $\mathbf{y}_{n-1}$, the overall network state, denoted by $\mathbf{s}_n$, is defined as

$$\mathbf{s}_n = [u(n), \ldots, u(n-K+1), x_1(n), \ldots, x_N(n), y(n-1), \ldots, y(n-L)]^{\mathrm{T}}, \tag{12.27}$$

while the internal unit dynamics are updated according to

$$\mathbf{x}_n = \Phi\big(\mathbf{W}_{ip}\mathbf{u}_n + \mathbf{W}_{in}\mathbf{x}_{n-1} + \mathbf{W}_b\mathbf{y}_{n-1}\big), \tag{12.28}$$

where $\Phi(\cdot)$ here is a quaternion-valued nonlinear activation of the neurons within the reservoir. The echo state property plays a key role in ensuring adequate operation of the dynamical reservoir of ESNs, which means that the internal neurons $\mathbf{x}_n$ within the dynamical reservoir of ESNs should display systematic variants of the exciting external signal $\mathbf{u}_n$, working as its echo functions. A practical procedure to ensure well-prepared RNNs with echo state property adopts a two-step operation on $\mathbf{W}_{in}$: (1) an internal weight matrix $\mathbf{W}_{in}$ is randomly drawn from a uniform distribution over a symmetric interval, (2) $\mathbf{W}_{in}$ is rescaled by $\mathbf{W}_{in} \leftarrow \mathbf{W}_{in}/|\lambda_{\max}|$, where $|\lambda_{\max}|$ is the largest absolute eigenvalue of $\mathbf{W}_{in}$ (spectral radius). In this way, the initialization of input and feedback connections stored in $\mathbf{W}_{ip}$ and $\mathbf{W}_b$ can be arbitrary [24,25]. To find recent advances in echo state property analysis, readers are advised to refer to [45–47]. The output of a nonlinear output mapping of ESNs is given by

$$y(n) = \Phi\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big), \tag{12.29}$$

where $\mathbf{w}_n$ is the weight vector associated with the output layer, for which the value is updated by minimizing suitable cost functions. The cost function chosen here is the instantaneous squared error, defined as

$$E(n) = \frac{1}{2}|e(n)|^2 = \frac{1}{2}e(n)e^*(n), \tag{12.30}$$

where $d(n)$ is the desired (teaching) signal and $e(n)$ is the instantaneous output error, given by $e(n) = d(n) - y(n)$. Note that $E(n)$ in (12.30) is real-valued, dependent on both quaternion-valued $e(n)$ and $e^*(n)$, and for such functions, it has been shown by $\mathbb{HR}$ calculus that the maximum change of gradient lies in the direction of the conjugate gradient, which is in accordance with the corresponding solution in $\mathbb{C}$ [43]. In this way, the minimization of $E(n)$ through gradient descent is given by

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \nabla_{\mathbf{w}^*} E(n), \tag{12.31}$$

where μ is the step size, a small positive constant. Using the chain rule, the gradient $\nabla_{\mathbf{w}^*} E(n)$ in (12.31) can be derived as

$$\begin{aligned}\nabla_{\mathbf{w}^*} E(n) &= \frac{1}{2}\frac{\partial |e(n)|^2}{\partial \mathbf{w}_n^*} \\ &= \frac{1}{2}\left(e(n)\frac{\partial e^*(n)}{\partial \mathbf{w}_n^*} + \frac{\partial e(n)}{\partial \mathbf{w}_n^*}e^*(n)\right),\end{aligned} \tag{12.32}$$

where

$$e(n) = d(n) - y(n) = d(n) - \Phi\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big). \tag{12.33}$$

Note that due to the noncommutativity of the quaternion product, the partial derivatives in (12.32) cannot be swapped with $e(n)$ and $e^*(n)$. Using the fact that $\Phi^*(q) = \Phi(q^*)$ and $(q_1q_2)^* = q_2^*q_1^*$, the output error conjugate $e^*(n)$ can be derived as

$$\begin{aligned} e^*(n) &= d^*(n) - \Phi^*\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big) \\ &= d^*(n) - \Phi\big((\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)^*\big) \\ &= d^*(n) - \Phi\big(\mathbf{s}_n^{\mathrm{H}}\mathbf{w}_n^*\big).\end{aligned} \tag{12.34}$$

Substituting it into the first gradient on the right-hand side of (12.32), we have

$$\begin{aligned}\frac{\partial e^*(n)}{\partial \mathbf{w}_n^*} &= \frac{\partial\big(d^*(n) - \Phi(\mathbf{s}_n^{\mathrm{H}}\mathbf{w}_n^*)\big)}{\partial \mathbf{w}_n^*} \\ &= -\frac{\partial \Phi\big(\mathbf{s}_n^{\mathrm{H}}\mathbf{w}_n^*\big)}{\partial \mathbf{w}_n^*} \\ &= -\Phi'\big(\mathbf{s}_n^{\mathrm{H}}\mathbf{w}_n^*\big)\frac{\partial\big(\mathbf{s}_n^{\mathrm{H}}\mathbf{w}_n^*\big)}{\partial \mathbf{w}_n^*} \\ &= -\Phi'^*\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\mathbf{s}_n^*,\end{aligned} \tag{12.35}$$

where the last step is achieved by using the $\mathbb{HR}$-derivative introduced in (12.17) in Section 12.2.3. On the other hand,

$$\begin{aligned}\frac{\partial e(n)}{\partial \mathbf{w}_n^*} &= \frac{\partial\big(d(n) - \Phi(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)\big)}{\partial \mathbf{w}_n^*} \\ &= -\frac{\partial \Phi(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)}{\partial \mathbf{w}_n^*} \\ &= -\Phi'(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)\frac{\partial\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)}{\partial \mathbf{w}_n^*}. \end{aligned} \tag{12.36}$$

Using the $\mathbb{HR}^*$-derivative given in (12.18), we obtain

$$\frac{\partial\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)}{\partial \mathbf{w}_n^*} = -\frac{\mathbf{s}_n}{2} \tag{12.37}$$

and hence

$$\frac{\partial e(n)}{\partial \mathbf{w}_n^*} = \frac{\Phi'\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\mathbf{s}_n}{2}. \tag{12.38}$$

Finally, we arrive at the update of the weight vector $\mathbf{w}_n$ in the output layer, given by

$$\begin{aligned}\mathbf{w}_{n+1} &= \mathbf{w}_n - \mu\nabla_{\mathbf{w}^*}E(n) \\ &= \mathbf{w}_n + \mu\Big(e(n)\Phi'^*\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\mathbf{s}_n^* - \frac{1}{2}\Phi'\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\mathbf{s}_n e^*(n)\Big),\end{aligned} \tag{12.39}$$

where the factor $1/2$ in the cost function (12.30) is absorbed into the step size μ.

12.3.2 AUGMENTED QESNS

In order to make QESNs optimal for general quaternion signals (both second-order circular and noncircular), we use recent developments in augmented quaternion statistics, as described in Section 12.2.2, to incorporate the widely linear model into the QESN architecture, which gives the augmented version of QESNs (AQESNs). This means that the network state should be "augmented" in such a way as

$$\mathbf{s}_n^a = \big[\mathbf{u}_n, \mathbf{u}_n^{\imath}, \mathbf{u}_n^{\jmath}, \mathbf{u}_n^{\kappa}, \mathbf{x}_n^a, \mathbf{y}_n\big]^{\mathrm{T}}. \tag{12.40}$$

This augmented network state design is generally applicable for not only the architecture of QESNs, but also quaternion-valued extensions of other types of RNNs. Since the input weights of QESNs contained in the matrix $\mathbf{W}_{ip}$ are randomly chosen prior to training, we can use three other matrices $\mathbf{W}_{ip1}$, $\mathbf{W}_{ip2}$ and $\mathbf{W}_{ip3}$ to initialize the weights associated with the input involutions $\mathbf{u}_n^{\imath}$, $\mathbf{u}_n^{\jmath}$, $\mathbf{u}_n^{\kappa}$. To this point, the update of the internal state dynamics within the AQESNs is given by

$$\mathbf{x}_n^a = \Phi\big(\mathbf{W}_{ip}\mathbf{u}_n + \mathbf{W}_{ip1}\mathbf{u}_n^{\imath} + \mathbf{W}_{ip2}\mathbf{u}_n^{\jmath} + \mathbf{W}_{ip3}\mathbf{u}_n^{\kappa} + \mathbf{W}_{in}\mathbf{x}_{n-1} + \mathbf{W}_b\mathbf{y}_{n-1}\big). \tag{12.41}$$

Due to the specific properties of the ESN output layer, the output $y(n)$ is now governed by an asymmetric version of the quaternion widely linear model in (12.14) to yield

$$\begin{aligned} net(n) &= \mathbf{a}_n^{\mathrm{T}}\mathbf{v}_n + \mathbf{b}_n^{\mathrm{T}}\mathbf{u}_n^{\imath} + \mathbf{c}_n^{\mathrm{T}}\mathbf{u}_n^{J} + \mathbf{d}_n^{\mathrm{T}}\mathbf{u}_n^{\kappa}, \\ y(n) &= \Phi\big(net(n)\big), \end{aligned} \tag{12.42}$$

where $\mathbf{v}_n = [u(n), \dots, u(n-K+1), x_1^a(n), \dots x_N^a(n), y(n-1), \dots, y(n-L)]^{\mathrm{T}}$ is a subset of the augmented network state $\mathbf{s}_n^a$ and has the same dimension $(K+N+L)$ as the state vector $\mathbf{s}_n$ within standard ESNs; however, the internal state dynamics are updated using (12.41). The recursion of the output weight vectors $\{\mathbf{a}_n, \mathbf{b}_n, \mathbf{c}_n, \mathbf{d}_n\}$ are made gradient adaptive according to

$$\begin{aligned} \mathbf{a}_{n+1} &= \mathbf{a}_n - \mu\nabla_{\mathbf{a}^*}E(n), \\ \mathbf{b}_{n+1} &= \mathbf{b}_n - \mu\nabla_{\mathbf{b}^*}E(n), \\ \mathbf{c}_{n+1} &= \mathbf{c}_n - \mu\nabla_{\mathbf{c}^*}E(n), \\ \mathbf{d}_{n+1} &= \mathbf{d}_n - \mu\nabla_{\mathbf{d}^*}E(n). \end{aligned} \tag{12.43}$$

We notice that the gradient $\nabla_{\mathbf{a}^*}E(n)$ in (12.43) and its counterpart $\nabla_{\mathbf{s}^*}E(n)$ in (12.31) are equivalent, so that they have the same dimension, and hence

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \mu\Big(e(n)\Phi'^*(net(n))\mathbf{v}_n^* - \frac{1}{2}\Phi'(net(n))\mathbf{v}_n\Big). \tag{12.44}$$

The error gradient $\nabla_{\mathbf{b}^*}E(n)$ for the weight vector $\mathbf{b}_n$ corresponding to the $\imath$-involution $\mathbf{u}_n^{\imath}$ of the input $\mathbf{u}$ is given by

$$\nabla_{\mathbf{b}^*}E(n) = \frac{1}{2}\left(e(n)\frac{\partial e^*(n)}{\partial \mathbf{b}_n^*} + \frac{\partial e(n)}{\partial \mathbf{b}_n^*}e^*(n)\right), \tag{12.45}$$

where

$$\begin{aligned} \frac{\partial e^*(n)}{\partial \mathbf{b}_n^*} &= -\frac{\partial \Phi\big(net^*(n)\big)}{\partial \mathbf{b}_n^*} \\ &= -\Phi'^*\big(net(n)\big)\frac{\partial net^*(n)}{\partial \mathbf{b}_n^*} \\ &= -\Phi'^*\big(net(n)\big)\mathbf{u}_n^{\imath *} \end{aligned} \tag{12.46}$$

and

$$\begin{aligned} \frac{\partial e(n)}{\partial \mathbf{b}_n^*} &= -\frac{\partial \Phi\big(net(n)\big)}{\partial \mathbf{b}_n^*} \\ &= -\Phi'\big(net(n)\big)\frac{\partial net(n)}{\partial \mathbf{b}_n^*} \\ &= \frac{1}{2}\Phi'\big(net(n)\big)\mathbf{u}_n^{\imath}. \end{aligned} \tag{12.47}$$

Substituting the partial derivatives in (12.46) and (12.47) into the error gradient $\nabla_{\mathbf{b}^*}E(n)$ given in (12.45) yields

$$\mathbf{b}_{n+1} = \mathbf{b}_n + \mu\Big(e(n)\Phi'^{*}\big(net(n)\big)\mathbf{u}_n^{\imath *} - \frac{1}{2}\Phi'\big(net(n)\big)\mathbf{u}_n^{\imath}e^*(n)\Big).$$

Proceeding in a similar manner, the weight updates for $\mathbf{c}_n$ and $\mathbf{d}_n$ are found to be

$$\mathbf{c}_{n+1}{=}\mathbf{c}_n + \mu\Big(e(n)\Phi'^{*}\big(net(n)\big)\mathbf{u}_n^{J*} - \frac{1}{2}\Phi'\big(net(n)\big)\mathbf{u}_n^{J}e^*(n)\Big) \tag{12.48}$$

and

$$\mathbf{d}_{n+1}{=}\mathbf{d}_n + \mu\Big(e(n)\Phi'^{*}\big(net(n)\big)\mathbf{u}_n^{\kappa *} - \frac{1}{2}\Phi'\big(net(n)\big)\mathbf{u}_n^{\kappa}e^*(n)\Big). \tag{12.49}$$

For convenience, the final weight update of the gradient decent algorithm used to train the output layer of the AQESNs can be written in an augmented form using (12.40) as

$$\mathbf{w}_{n+1}^a = \mathbf{w}_n^a + \mu\Big(e(n)\Phi'^{*}\big(net(n)\big)\mathbf{s}_n^{a*} - \frac{1}{2}\Phi'\big(net(n)\big)\mathbf{s}_n^{a}e^*(n)\Big), \tag{12.50}$$

where $\mathbf{w}_n^a = [\mathbf{a}_n^{\mathrm{T}}, \mathbf{b}_n^{\mathrm{T}}, \mathbf{c}_n^{\mathrm{T}}, \mathbf{d}_n^{\mathrm{T}}]^{\mathrm{T}}$ is the augmented weight vector and $net(n) = \mathbf{w}_n^{a\mathrm{T}}\mathbf{s}_n^a$.

12.3.3 STABILITY ANALYSIS OF QESNS AND AQESNS

In order to ensure satisfactory performances of both QESNs and AQESNs, we now address the set bounds on the values of the step size μ to guarantee their stability. Following the approach from [2], we employ the mean square convergence criterion, given by

$$E\Big[|\bar{e}(n)|^2\Big] < E\Big[|\tilde{e}(n)|^2\Big], \tag{12.51}$$

where $\bar{e}(n)$ and $\tilde{e}(n)$ are respectively the *a posteriori* and the *a priori* output error, defined as

$$\begin{aligned}\bar{e}(n) &= d(n) - \Phi\big(\mathbf{w}_{n+1}^{\mathrm{T}}\mathbf{s}_n\big),\\ \tilde{e}(n) &= d(n) - \Phi\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big).\end{aligned} \tag{12.52}$$

The error terms $\bar{e}(n)$ and $\tilde{e}(n)$ in (12.52) can be related by the first-order Taylor series expansion (TSE) as [48]

$$\begin{aligned}|\bar{e}(n)|^2 &= |\tilde{e}(n)|^2 + \frac{\partial|\tilde{e}(n)|^2}{\partial\mathbf{w}_n^{\mathrm{T}}}\Delta\mathbf{w}_n + \frac{\partial|\tilde{e}(n)|^2}{\partial\mathbf{w}_n^{\imath\mathrm{T}}}\Delta\mathbf{w}_n^{\imath} + \frac{\partial|\tilde{e}(n)|^2}{\partial\mathbf{w}_n^{J\mathrm{T}}}\Delta\mathbf{w}_n^{J} + \frac{\partial|\tilde{e}(n)|^2}{\partial\mathbf{w}_n^{\kappa\mathrm{T}}}\Delta\mathbf{w}_n^{\kappa}\\ &= |\tilde{e}(n)|^2 + 4\Re\left(\frac{\partial|\tilde{e}(n)|^2}{\partial\mathbf{w}_n^{\mathrm{T}}}\Delta\mathbf{w}_n\right).\end{aligned} \tag{12.53}$$

To simplify the derivation of (12.53), notice that $\Re(q_1 q_2) = \Re((q_1 q_2)^*) = \Re(q_2^* q_1^*)$ for any pair $\{q_1, q_2\} \in \mathbb{H}$ to give

$$|\bar{e}(n)|^2 = |\tilde{e}(n)|^2 + 4\Re\left(\Delta\mathbf{w}_n^{\mathrm{H}} \frac{\partial|\tilde{e}(n)|^2}{\partial\mathbf{w}_n^*}\right), \tag{12.54}$$

where the partial derivative $\partial|\tilde{e}(n)|^2/\partial\mathbf{w}_n^*$ is effectively the gradient of the cost function with respect to $\mathbf{w}_n^*$, given in (12.32). The term $\Delta\mathbf{w}_n^{\mathrm{H}}$ can be expressed by using (12.39) as

$$\begin{aligned}
\Delta\mathbf{w}_n^{\mathrm{H}} &= \big(\mathbf{w}_{n+1} - \mathbf{w}_n\big)^{\mathrm{H}} \\
&= \mu\Big(\tilde{e}(n)\Phi'^*\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\mathbf{s}_n^* - \frac{1}{2}\Phi'\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\mathbf{s}_n\tilde{e}^*(n)\Big)^{\mathrm{H}} \\
&= \mu\Big(\mathbf{s}_n^{\mathrm{T}}\Phi'\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\tilde{e}^*(n) - \frac{1}{2}\tilde{e}(n)\mathbf{s}_n^{\mathrm{H}}\Phi'^*\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\Big).
\end{aligned} \tag{12.55}$$

Substituting (12.32) and (12.55) into the TSE in (12.54) yields

$$\begin{aligned}
|\bar{e}(n)|^2 &= |\tilde{e}(n)|^2 - 4\mu\Re\Big(\big(\mathbf{s}_n^{\mathrm{T}}\Phi'(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)\tilde{e}^*(n) - \frac{1}{2}\tilde{e}(k)\mathbf{s}_n^{\mathrm{H}}\Phi'^*(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)\big) \\
&\quad\cdot \big(\tilde{e}(n)\Phi'^*(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)\mathbf{s}_n^* - \frac{1}{2}\Phi'(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)\mathbf{s}_n\tilde{e}^*(n)\big)\Big) \\
&= |\tilde{e}(n)|^2\Big(1{-}5\mu|\Phi'(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)|^2\|\mathbf{s}_n\|_2^2|\Big) \\
&\quad + 4\mu\Re\Big(\mathbf{s}_n^{\mathrm{T}}\Phi'\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\tilde{e}^*(n)\Phi'\big(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n\big)\mathbf{s}_n\tilde{e}^*(n)\Big).
\end{aligned} \tag{12.56}$$

Given that the term $\Re(\cdot)$ is negligible, upon applying the statistical expectation operator on both sides of (12.56), we have

$$E\Big[|\bar{e}(n)|^2\Big] \;=\; E\Big[|\tilde{e}(n)|^2\big(1 - 5\mu|\Phi'(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)|^2\|\mathbf{s}_n\|_2^2|\big)\Big]. \tag{12.57}$$

By using the standard statistical independence assumption between $\tilde{e}(n)$ and $\mathbf{s}_n$, we have

$$E\Big[|\bar{e}(n)|^2\Big] \;=\; E\Big[|\tilde{e}(n)|^2\Big]E\Big[1 - 5\mu|\Phi'(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)|^2\|\mathbf{s}_n\|_2^2\Big]. \tag{12.58}$$

Therefore, the convergence condition in (12.51) is satisfied for

$$0 < E\Big[5\mu|\Phi'(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)|^2\|\mathbf{s}_n\|_2^2\Big] < 1. \tag{12.59}$$

Solving for μ, we obtain the range of the step size for QESNs to converge as

$$0 < \mu < \frac{1}{E\Big[5|\Phi'(\mathbf{w}_n^{\mathrm{T}}\mathbf{s}_n)|^2\|\mathbf{s}_n\|_2^2\Big]}. \tag{12.60}$$

Similarly, the bounds on μ to ensure the convergence of AQESNs are given by

$$0 < \mu < \frac{1}{E\left[5|\Phi'(\mathbf{w}_n^{a\mathrm{T}}\mathbf{s}_n^a)|^2\|\mathbf{s}_n^a\|_2^2\right]}. \tag{12.61}$$

The upper bounds of both (12.60) and (12.61) are governed by the expected value of the network state vector and the gradient of the fully quaternion nonlinearity. However, we should note that the upper bound of μ for the AQESNs in (12.61) is smaller than that of the standard QESNs in (12.60), due to the larger size of the augmented network state vector $\mathbf{s}_n^a$.

12.4 SIMULATIONS

To verify the suitability of QESNs to cater for 3D and 4D nonlinear processes and our idea of exploiting the noncircularity of multidimensional data through the widely linear model, simulations in a prediction setting were performed for both benchmark synthetic second-order circular (proper) and noncircular (improper) 3D and 4D signals and real-world noncircular, nonlinear and nonstationary 3D body motion and wind data. In order to implement the adequate buildup of QESNs, which are guaranteed to maintain the echo state property, the randomly selected input, internal and feedback weights $\mathbf{W}_{ip}$, $\mathbf{W}_{in}$ and $\mathbf{W}_b$ were generated from a uniform distribution in range $[-1, 1]$, and the spectral radius of $\mathbf{W}_{in}$ was 0.8. The internal weights connections contained in $\mathbf{W}_{in}$ were sparse with 5% interconnectivity, so that the reservoir contained many loosely coupled subsystems, and a rich variety of dynamics of different internal units can be achieved [24,25]. The fully quaternion $\tanh(q)$ function was employed as the activation function of QESNs. The step size μ of the gradient descent algorithms used to train the output layers of QESNs and augmented (widely linear) QESNs (AQESNs) was set at $\mu = 0.02$. To assess the quantitative performance of learning algorithms, it is convenient to use the prediction gain $\mathrm{R_p}$, defined as

$$\mathrm{R_p} = 10\log_{10}\frac{\hat{\sigma}_q^2}{\hat{\sigma}_e^2}, \tag{12.62}$$

which is a logarithmic ratio between the estimated signal variance $\hat{\sigma}_q^2$ and estimated prediction error variance $\hat{\sigma}_e^2$. Five input neurons and five output neurons were used and the different dynamical reservoir sizes of ESNs were investigated. Due to their random network preparation, all the simulation results were obtained by averaging 100 independent simulation trials.

In the first set of simulations, we considered a benchmark synthetic second-order circular (proper) 4D autoregressive AR(4) process, given by [2]

$$u(n) = 1.79u(n-1) - 1.85u(n-2) + 1.27u(n-3) - 0.41u(n-4) + q(n), \tag{12.63}$$

driven by a circular quaternion white Gaussian noise (QWGN) $q(n) = q_r(n) + \imath q_\imath(n) + \jmath q_\jmath(n) + \kappa q_\kappa(n)$, where $q_r(n)$, $q_\imath(n)$, $q_\jmath(n)$ and $q_\kappa(n)$ are independent realizations of real-valued WGN $\sim \mathcal{N}(0, 1)$.

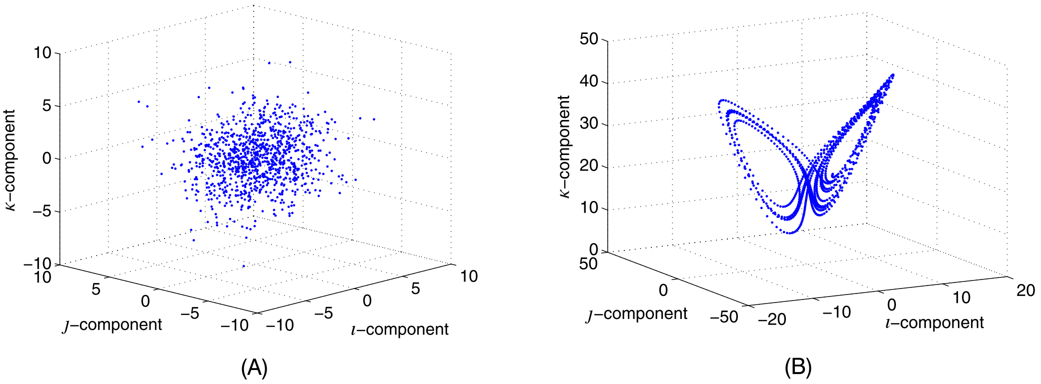

FIGURE 12.2

Geometric view of circularity via a 3D scatter diagram. (A) Circular 4D AR(4) signal. (B) Noncircular and nonlinear 3D chaotic Lorenz signal. For illustration purpose, only the ı-, ȷ- and κ-components of the quaternion AR(4) signal are provided.

Table 12.1 Comparison of prediction gains R_p of standard QESNs and AQESNs, as well as strictly linear QLMS and its augmented (widely linear) version AQLMS adaptive learning algorithms, for the benchmark signals

R_p [dB]	4D AR(4)	3D Lorenz
QLMS	3.42	5.24
AQLMS	3.38	10.02
QESNs	3.39	14.54
AQESNs	3.35	17.96

The benchmark noncircular and nonlinear signal considered was the chaotic 3D Lorenz attractor, governed by coupled partial differential equations [49]

$$\frac{\partial x}{\partial t} = \alpha(y - x), \ \frac{\partial y}{\partial t} = x(\rho - z) - y, \ \frac{\partial z}{\partial t} = xy - \beta z, \tag{12.64}$$

where $\alpha = 10$, $\rho = 28$ and $\beta = 8/3$, and it was constructed as a pure quaternion by using the three imaginary components.

Fig. 12.2 shows the 3D scatter plots of the considered benchmark signals, in which the 4D AR(4) process had a rotation invariant distribution (circular), while the 3D Lorenz signal possessed an obvious noncircular nature. Table 12.1 compares the prediction gains R_p of QESNs and AQESNs with 100 internal neurons in the dynamical reservoir, as well as strictly linear quaternion least mean square (QLMS) and its widely linear version augmented QLMS (AQLMS) adaptive learning algorithms [16, 37,43], for the benchmark signals considered. For the strictly linear and circular 4D AR(4) signal,

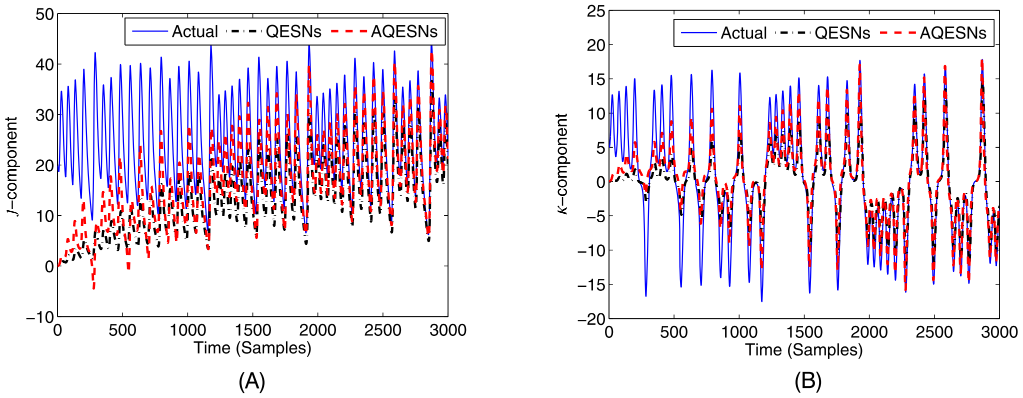

FIGURE 12.3

One-step ahead prediction of the noncircular and nonlinear 3D Lorenz signal using QESNs and AQESNs. Amplitudes of the actual and predicted signals are plotted. (A) ȷ-component. (B) κ-component.

the performance of AQESNs was comparable but slightly worse than that of QESNs. This is because the widely linear model–based input layer of AQESNs simplifies into the strictly linear one used in QESNs, as the weights associated with the involutions of the input vector, that is, $\{\mathbf{b}, \mathbf{c}, \mathbf{d}\}$ in (12.41), vanish theoretically; however, in practice, small weight deviations from zero arise from the employed learning algorithm. For the noncircular 3D Lorenz signal, there was a significant overall performance improvement in terms of the prediction gain R_p when the widely linear model–based AQESNs were employed as compared with the strictly linear QESNs. This is due to their faster and more accurate tracking capability of the signal dynamics arising from their natural exploitation of data noncircularity, as indicated by Fig. 12.3. The main motivation to design quaternion-valued nonlinear learning algorithms with enhanced modeling capacities for nonlinear dynamic systems, as compared with the strictly linear learning algorithms, has been also justified by observing the higher prediction gains achieved by QESNs and AQESNs than those of QLMS and AQLMS algorithms, on the nonlinear and noncircular 3D Lorenz signal. As expected, for strictly linear and circular 4D AR(4) process, all the considered learning algorithms yielded comparable prediction results.

In the next set of simulations, we investigated the usefulness of the QESNs and AQENs for a real-world 3D noncircular and nonstationary body motion tracking experiment. The 3D motion data were recorded using the XSense MTx 3-DOF orientation tracker, placed on the left and right hands, the left and right arms and the waist of an athlete performing Tai Chi movements. The movement of the left arm was used as a pure quaternion input, and QESNs and AQESNs with 200 internal neurons were employed for this experiment. Fig. 12.4A–C show that along all the three dimensions, AQESNs were able to track the left arm dynamics of the athlete more quickly and more accurately than their strictly linear counterparts, which is further supported by the evolution of their mean square errors (MSEs), shown in Fig. 12.4D.

Wind forecasting at short scales plays an important role in renewable energy, smart grids, air pollution modeling and aviation safety [50]. To further demonstrate the modeling ability of the designed quaternion-valued nonlinear learning systems in the fusion of heterogeneous data sources, a 4D wind model was considered in such a way that the 3D noncircular, nonlinear and nonstationary wind speed

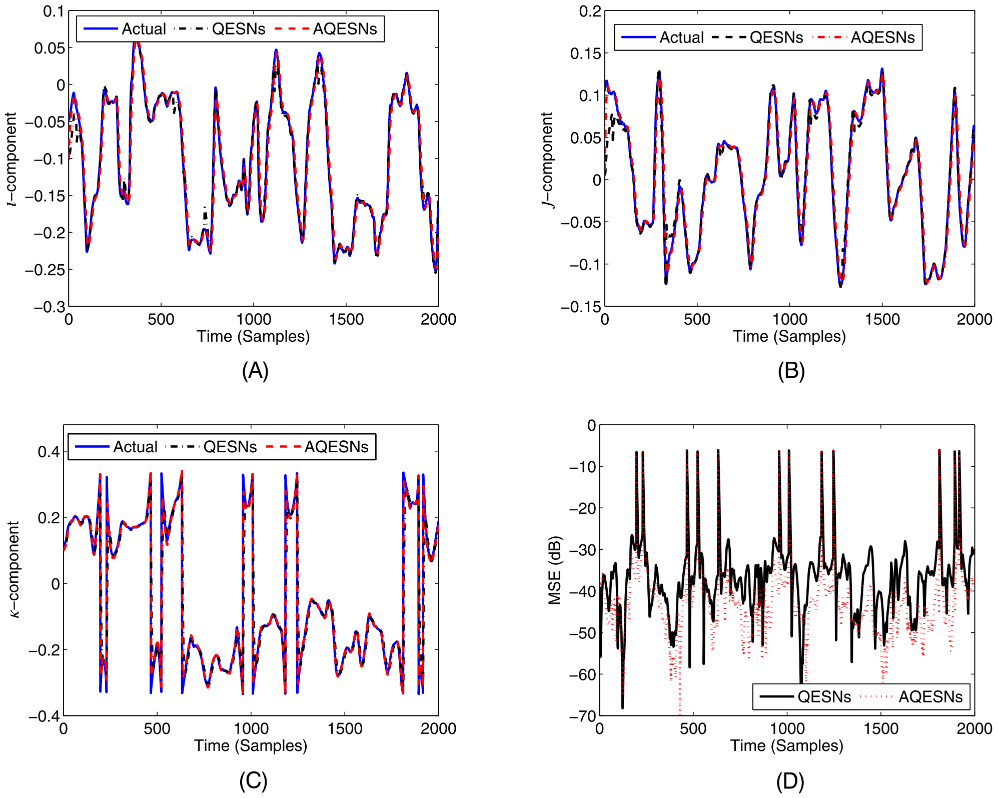

FIGURE 12.4

Tracking capabilities and MSE performances, evaluated over all the three dimensions, of QESNs and AQESNs on a one-step ahead prediction of the 3D left arm motion of an athlete performing Tai Chi movements. (A) $\imath$-component. (B) $\jmath$-component. (C) κ-component. (D) MSE (dB).

data in east, north and vertical directions constituted the vector part of the quaternion, and the air temperature was used as a scalar (real) part. Note that the validity of incorporating air temperature in wind forecasting has been justified by the fact that the air temperature has a high degree of correlation with the wind speed [51]. Fig. 12.5A shows the overall prediction performance over the three dimensions of the wind speed data for QESNs with 100 internal neurons, where an improved accuracy was achieved by incorporating the widely linear model into QESNs. It is interesting to note that the widely linear model enabled AQESNs to more closely follow the highly nonstationary trend of the air temperature, as indicated by Fig. 12.5B. From Fig. 12.5A and Fig. 12.5B, we also observe that the strictly linear QLMS learning algorithm, and even its augmented version AQLMS, were not able to track the highly nonlinear dynamics of the heterogeneous 4D wind data. To further illustrate the robustness of the performance advantage of using widely linear model within QESNs, we compared the performances of both augmented and standard QESNs against the size of the dynamical reservoir, an important parameter that influences the performance of ESNs, as it reflects their universal approximation capability. Generally speaking, an ESN with a larger reservoir can learn the signal dynamics with higher accu-

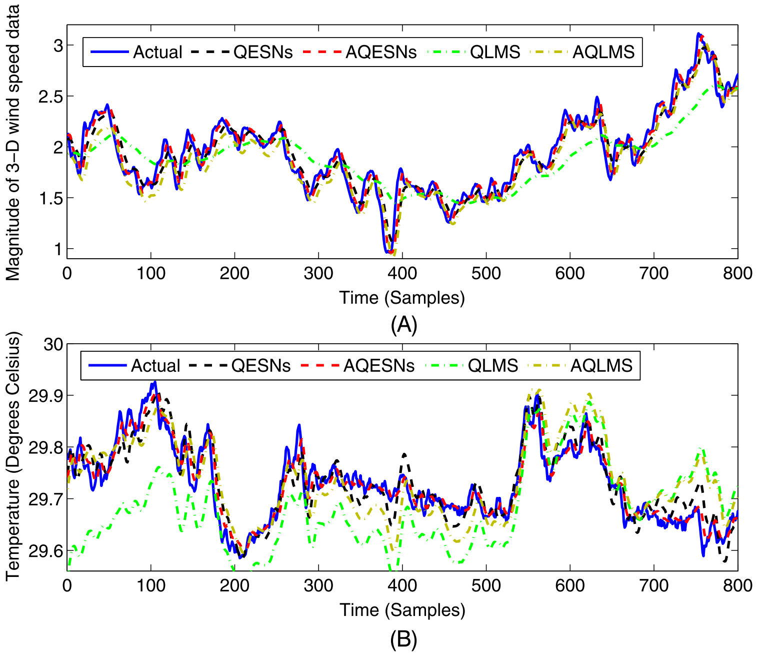

FIGURE 12.5

One-step ahead prediction of the 4D wind data using QESNs and AQESNs with 100 internal neurons, and strictly linear QLMS and AQLMS learning algorithms. (A) Magnitude of the 3D wind speed. (B) Air temperature.

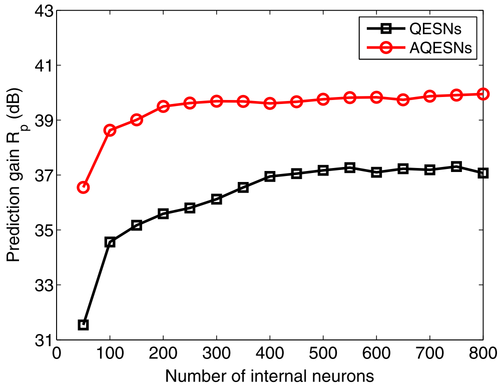

FIGURE 12.6

Performance comparison of QESNs and AQESNs over a range of reservoir sizes on one-step ahead prediction of the 4D wind data.

racy. This is confirmed in Fig. 12.6, where the performances of QESNs and AQESNs are investigated against the reservoir size. In all cases, the AQESNs outperformed their standard counterparts.

12.5 DISCUSSION AND CONCLUSION

We have introduced QESNs for nonlinear adaptive filtering of 3D and 4D processes. This is achieved by employing full quaternion nonlinear activation functions with local analyticity, allowing for first-order differentiability at the operating point to design quaternion-valued gradient descent algorithms. Unlike currently existing quaternion-valued RNNs, such as the fully connected quaternion RNNs and the quaternion Hopfield neural networks [52], with well-prepared sparse connected dynamical reservoir prior to training, QESNs require training only for the weights connecting the hidden layer and the readout neurons, significantly reduce computational complexity and, hence, enable the possibility to build up large-scale quaternion RNNs with acceptable computational costs. To further make QESNs optimal for the generality of quaternion-valued signals (both second-order circular and noncircular), the widely linear model has been incorporated into QESNs to introduce AQESNs. The performance advantage of exploiting data noncircularity through the widely linear model has been illustrated by simulations over a range of noncircular synthetic 3D and 4D signals and for real-world noncircular, nonlinear and nonstationary 3D body motion recordings and 4D wind data via a fusion of heterogeneous data sources.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China under Grants 61401094 and 61771124, the Natural Science Foundation of Jiangsu Province under Grant BK20140645 and the Fundamental Research Funds for the Central Universities under Grant 2242016K41050.

REFERENCES

[1] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals, and Systems 2 (4) (1989) 303–314.

[2] D.P. Mandic, J. Chambers, Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability, Wiley, New York, NY, USA, 2001.

[3] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks 1 (1) (1990) 4–27.

[4] S. Haykin, L. Li, Nonlinear adaptive prediction of nonstationary signals, IEEE Transactions on Signal Processing 43 (2) (1995) 526–535.

[5] S.A. Billings, C.F. Fung, Recurrent radial basis function networks for adaptive noise cancellation, Neural Networks 8 (2) (1995) 273–290.

[6] J.L. Elman, Finding structure in time, Cognitive Science 14 (2) (1990) 179–211.

[7] S.B. Choe, J.J. Faraway, Modeling head and hand orientation during motion using quaternions, Journal of Aerospace Engineering 113 (1) (2004) 186–192.

[8] C.F. Karney, Quaternions in molecular modeling, Journal of Molecular Graphics & Modelling 25 (5) (2007) 595–604.

[9] T.A. Ell, S.J. Sangwine, Hypercomplex Fourier transforms of color images, IEEE Transactions on Image Processing 16 (1) (2007) 22–35.

[10] A. Kolaman, O. Yadid-Pecht, Quaternion structural similarity: a new quality index for color images, IEEE Transactions on Image Processing 21 (4) (2012) 1526–1536.

[11] N. Le Bihan, J. Mars, Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing, Signal Processing 84 (7) (2004) 1177–1199.

[12] N. Le Bihan, S. Miron, J. Mars, MUSIC algorithm for vector-sensors array using biquaternions, IEEE Transactions on Signal Processing 55 (9) (2007) 4523–4533.

[13] A.M. Sabatini, Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing, IEEE Transactions on Aerospace and Electronic Systems 53 (7) (2006) 1346–1356.

[14] D. Choukkroun, I.Y. Bar-Itzhack, Y. Ohsman, Novel quaternion Kalman filter, IEEE Transactions on Aerospace and Electronic Systems 42 (1) (2006) 174–190.

[15] C. Jahanchahi, D.P. Mandic, A class of quaternion Kalman filters, IEEE Transactions on Neural Networks and Learning Systems 25 (3) (2014) 533–544.

[16] C. Cheong-Took, D.P. Mandic, The quaternion LMS algorithm for adaptive filtering of hypercomplex processes, IEEE Transactions on Signal Processing 57 (4) (2009) 1316–1327.

[17] J. Via, D.P. Palomar, L. Viela, I. Santamaria, Quaternion ICA from second-order statistics, IEEE Transactions on Signal Processing 59 (4) (2011) 1586–1600.

[18] A. Sudbery, Quaternionic analysis, Mathematical Proceedings of the Cambridge Philosophical Society 85 (2) (1979) 199–225.

[19] T. Isokawa, H. Nishuimura, N. Kamiura, N. Matsui, Associative memory in quaternionic Hopfield neural network, International Journal of Neural Systems 18 (2) (2008) 135–145.

[20] B.C. Ujang, C. Cheong-Took, D.P. Mandic, Split quaternion nonlinear adaptive filtering, Neural Networks 23 (3) (2010) 426–434.

[21] S. De Leo, P. Rotelli, Quaternion analyticity, Applied Mathematics Letters 16 (7) (2003) 1077–1081.

[22] B.C. Ujang, C. Cheong-Took, D.P. Mandic, Quaternion-valued nonlinear adaptive filtering, IEEE Transactions on Neural Networks 22 (8) (2011) 1193–1206.

[23] B.C. Ujang, C. Cheong-Took, D.P. Mandic, On quaternion analyticity: enabling quaternion-valued nonlinear adaptive filtering in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Kyoto, Japan, March 25–March 30, 2012, pp. 2117–2120.

[24] H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication, Science 304 (5667) (2004) 78–80.

[25] M. Lukosevicius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Computer Science Review 3 (3) (2009) 127–149.

[26] I.B. Yildiz, H. Jaeger, S.J. Kiebel, Re-visiting the echo state property, Neural Networks 35 (2012) 1–9.

[27] B. Picinbono, On noncircularity, IEEE Transactions on Signal Processing 42 (12) (1994) 3473–3483.

[28] B. Picinbono, P. Chevailer, Widely linear estimation with complex data, IEEE Transactions on Signal Processing 43 (8) (1995) 2030–2033.

[29] D.P. Mandic, S.L. Goh, Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models, Wiley, New York, NY, USA, 2009.

[30] P.J. Schreier, L.L. Scharf, Statistical Signal Processing of Complex-Valued Data: The Theory of Improper and Noncircular Signals, Cambridge University Press, Cambridge, UK, 2010.

[31] L. Anttila, M. Valkama, M. Renfors, Circularity-based I/Q imbalance compensation in wideband direct-conversion receivers, IEEE Transactions on Vehicular Technology 57 (4) (2008) 2099–2113.

[32] Y. Xia, S.C. Douglas, D.P. Mandic, Adaptive frequency estimation in smart grid applications: exploiting noncircularity and widely linear adaptive estimators, IEEE Signal Processing Magazine 29 (5) (2012) 44–54.

[33] Y. Xia, D.P. Mandic, Complementary mean square analysis of augmented CLMS for second-order noncircular Gaussian signals, IEEE Signal Processing Letters 24 (9) (2017) 1413–1417.

[34] N.N. Vakhania, Random vectors with values in quaternion Hilbert spaces, Theory of Prabability & Its Application 43 (1) (1999) 99–115.

[35] P.O. Amblard, N. Le Bihan, On properness of quaternion valued random variables, in: Proceedings of the Sixth IMA International Conference on Mathematics in Signal Processing, Cirenceister, UK, December 14–December 16, 2004, pp. 23–26.

[36] J. Via, D. Ramirez, I. Santamaria, Properness and widely linear processing of quaternion random vectors, IEEE Transactions on Information Theory 56 (7) (2010) 3502–3515.

[37] C. Cheong-Took, D.P. Mandic, A quaternion widely linear adaptive filter, IEEE Transactions on Signal Processing 58 (8) (2010) 4427–4431.

[38] C. Cheong-Took, D.P. Mandic, Augmented second-order statistics of quaternion random signals, Signal Processing 91 (2) (2011) 214–224.

[39] J. Navarro-Moreno, R.M. Fernandez-Alcala, J.C. Ruiz-Molina, A quaternion widely linear series expansion and its applications, IEEE Signal Processing Letters 19 (12) (2012) 868–871.

[40] Y. Xia, C. Jahanchahi, T. Nitta, D.P. Mandic, Performance bounds of quaternion estimators, IEEE Transactions on Neural Networks and Learning Systems 26 (12) (2015) 3287–3292.

[41] Y. Xia, B. Jelfs, M.M. Van Hulle, J.C. Principe, D.P. Mandic, An augmented echo state networks for nonlinear adaptive filtering of complex noncircular signals, IEEE Transactions on Neural Networks 22 (1) (2011) 74–83.

[42] Y. Xia, C. Jahanchahi, D.P. Mandic, Quaternion-valued echo state networks, IEEE Transactions on Neural Networks and Learning Systems 26 (4) (2015) 663–673.

[43] D.P. Mandic, C. Jahanchahi, C. Cheong-Took, A quaternion gradient operator and its applications, IEEE Signal Processing Letters 18 (1) (2011) 47–50.

[44] C.A. Deavours, The quaternion calculus, American Mathematical Monthly 80 (9) (1973) 995–1008.

[45] M. Buehner, P. Young, A tighter bound for the echo state property, IEEE Transactions on Neural Networks 17 (3) (2006) 820–824.

[46] M.C. Ozturk, D. Xu, J.C. Principe, Analysis and design of echo state networks, Neural Computations 19 (1) (2007) 111–138.

[47] B. Zhang, D.J. Miller, Y. Wang, Nonlinear system modeling with random matrices: echo state networks revisited, IEEE Transactions on Neural Networks and Learning Systems 23 (1) (2012) 175–182.

[48] E. Soria-Olivas, J. Calpe-Maravilla, J.F. Guerrero-Martinez, M. Martinez-Sober, J. Espi-Lopez, An easy demonstration of the optimum value of the adaptation constant in the LMS algorithm [FIR filter theory], IEEE Transactions on Education 41 (1) (1998) 81.

[49] S.H. Stogatz, Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering, Studies in Nonlinearity, Westview, Boulder, CO, USA, 2001.

[50] J.F. Manwell, J.G. Mcgowan, A.L. Rogers, Wind Energy Explained: Theory, Design and Application, Wiley, West Sussex, UK, 2002.

[51] C. Cheong-Took, G. Strbac, A. Aihara, D.P. Mandic, Quaternion-valued short-term joint forecasting of three-dimensional wind and atmospheric parameters, Renewable Energy 36 (6) (2011) 1754–1760.

[52] M. Kobayashi, Fixed points of split quaternionic Hopfield neural networks, Signal Processing 136 (2017) 38–42.