Quaternion-Valued Echo State Networks

Yili Xia, *Member, IEEE*, Cyrus Jahanchahi, and Danilo P. Mandic, *Fellow, IEEE*

***Abstract*— Quaternion-valued echo state networks (QESNs) are introduced to cater for 3-D and 4-D processes, such as those observed in the context of renewable energy (3-D wind modeling) and human centered computing (3-D inertial body sensors). The introduction of QESNs is made possible by the recent emergence of quaternion nonlinear activation functions with local analytic properties, required by nonlinear gradient descent training algorithms. To make QENSs second-order optimal for the generality of quaternion signals (both circular and noncircular), we employ augmented quaternion statistics to introduce widely linear QESNs. To that end, the standard widely linear model is modified so as to suit the properties of dynamical reservoir, typically realized by recurrent neural networks. This allows for a full exploitation of second-order information in the data, contained both in the covariance and pseudocovariances, and a rigorous account of second-order noncircularity (improperness), and the corresponding power mismatch and coupling between the data components. Simulations in the prediction setting on both benchmark circular and noncircular signals and on noncircular real-world 3-D body motion data support the analysis.**

***Index Terms*— Augmented quaternion statistics, echo state networks (ESNs), second-order noncircularity, widely linear model.**

I. Introduction

RECURRENT neural networks (RNNs) are widely used for processing nonlinear and nonstationary signals, due to their ability to represent highly nonlinear dynamical systems, attractor dynamics, and long impulse responses [1], [2]. With the emergence of multidimensional sensors, several important RNN architectures have been extended to the complex domain $\mathbb{C}$, also catering for real-world bivariate signals. Examples include coherent neural networks for sensorimotor systems [3], widely linear complex RNNs for signal prediction [4], sonar signal prediction and image enhancement by multivalued neurons [5], grayscale image processing by complex-valued multistate neural associative memory [6], and geometric figure transformation via complex-valued multilayer networks [7].

Recent progress in sensing technology has made possible the recording from data sources, which are 3-D and 4-D, such as measurements from inertial body sensors and ultrasonic anemometers. These measurements can be represented as vectors in $\mathbb{R}^3$ and $\mathbb{R}^4$, however, vector algebra is not a division algebra and suffers from mathematical deficiencies when modeling orientation and rotation (gimbal lock). On the other hand, the quaternion domain $\mathbb{H}$ offers a convenient and unified means to process 3-D and 4-D signals. Quaternions have found application in computer graphics [8], molecular modeling [9], color image processing [10], 3-D polarized signal representation for vector-sensor array processing [11], [12], and modeling of 3-D wind signals with associated atmospheric parameters in renewable energy applications [13].

In the context of learning systems, quaternion approaches include both Kalman filtering [14], [15] and stochastic gradient algorithms [16]. However, in the context of nonlinear learning systems, quaternion-valued processing is still emerging, mainly due to the lack of analytic nonlinear functions in the quaternion domain $\mathbb{H}$. Namely, the very stringent Cauchy–Riemann–Fueter (CRF) conditions admit only linear functions and constants as globally analytic quaternion-valued functions [17]. This is a serious obstacle that prevents the standard nonlinear activation functions (tanh, logistic) from being the nonlinearities in nonlinear quaternion-valued estimation.

To partially overcome this issue, early approaches use a split quaternion function that treats each quaternion component separately (as a real channel) passed through a real smooth nonlinearity [18]–[20]. Although this may yield enhanced performance over vector-based algorithms, the noncommutativity aspect of the quaternion algebra is overlooked, thus prohibiting rigorous treatment of the cross-information and not exploiting full potential of quaternions. Recognizing that gradient-based learning, such as nonlinear gradient descent (NGD), and real-time recurrent learning (RTRL) [2], [21], [22], require gradient evaluation at a point, makes it possible to adopt a local alternative to the global CRF conditions, that is, the local analyticity condition (LAC) [23]. The work in [24] uses LAC to establish a class of neural networks in $\mathbb{H}$, which are a generic extensions 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 NGD (QNGD) for nonlinear FIR filters [24] and quaternion RTRL (QRTRL) for fully connected neural networks [25]. Another obvious obstacle, which hinders the development of quaternion RNNs is the high computational complexity associated with their training, as quaternion addition requires four real-valued additions whereas quaternion multiplication requires 16 real-valued multiplications and 12 real-valued additions.

To address these issues, in this paper, we generalize echo state networks (ESNs) [26]–[28] to enable the processing of hypercomplex 3-D and 4-D signals. The principle behind ESNs is to separate the RNN architecture into two constituent

Manuscript received April 23, 2013; revised February 11, 2014; accepted April 20, 2014. Date of publication May 21, 2014; date of current version March 16, 2015.

Y. Xia is with the School of Information Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: yili.xia06@gmail.com).

C. Jahanchahi and D. P. Mandic are with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: cyrus.jahanchahi05@imperial.ac.uk; d.mandic@imperial.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2320715

components: a recurrent dynamical reservoir in the hidden layer, and a readout neuron within the memoryless nonlinearity in the output layer, where the recurrent dynamical reservoir is built upon a randomly generated group of hidden neurons with a specified degree of recurrent connections, which satisfies the so-called echo state property to maintain stability [26]. This way, the high computational complexity of RNNs is significantly reduced due to the sparse connections among the hidden neurons. In addition, the training requirements are reduced to only the weights connecting the hidden layer and the readout neuron. To further equip the proposed quaternion ESNs (QESNs) with enhanced modeling capability for noncircular quaternion signals (rotation-dependent distribution), we employ recent developments in augmented quaternion statistics to incorporate the widely linear model [16], [29], [30] into QESNs, making them second-order optimal for the generality of quaternion signals (both circular and noncircular). The so introduced widely linear model for state space estimation is not limited to ESNs, but is also applicable to general RNN architectures. Simulations in the prediction setting on both benchmark circular and noncircular signals and on noncircular real world 3-D body motion data support the analysis.

The main contributions of this paper are therefore as follows.

1) By virtue of the sparsely connected dynamical reservoir, the training task is reduced to the weights linking the hidden layer and the readout neuron, so that the proposed QESNs significantly reduce the high computational complexity typically encountered by RNNs in $\mathbb{H}$.
2) The use of a fully quaternion nonlinearity in $\mathbb{H}$ instead of split quaternion functions within the derivation of the proposed QESNs training algorithms to preserve the cross-information within the data components.
3) The incorporation of widely linear model into QESN structures, so as to make them second-order optimal for the generality of quaternion signals; this design is also applicable to general RNNs in $\mathbb{H}$.

The rest of this paper is organized as follows. An overview of basic operations of quaternion algebra is provided in Section II. Section III reviews the augmented quaternion statistics and widely linear model in $\mathbb{H}$. The suitability of quaternion transcendental nonlinear functions as activation functions in $\mathbb{H}$ is addressed in Section IV. In Section V, the QESNs and its augmented (AQESNs) variant for the generality of quaternion signals (both second-order circular and noncircular) are derived, and simulation results are given in Section VI. Finally, Section VII concludes this paper.

II. Quaternion Algebra

The quaternion domain is a noncommutative extension of the complex-domain and provides a natural framework for processing 3-D and 4-D signals. A quaternion variable $q \in \mathbb{H}$ can be expressed as

$$q = q_r + \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa = Sq + Vq$$

and comprises 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 three imaginary components, $Vq = \Im(q) = \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa$. The imaginary units $\imath$, $\jmath$, and κ obey the following rules:

$$\imath\jmath = \kappa, \jmath\kappa = i, \kappa\imath = \jmath,$$
$$\imath^2 = \jmath^2 = \kappa^2 = \imath\jmath\kappa = -1.$$

Note that the quaternion multiplication is not commutative, that is, $\imath\jmath \neq \jmath\imath = -\kappa$. The product of quaternions, q_1 and $q_2 \in \mathbb{H}$, is given by

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

where the symbol $\cdot$ denotes the dot-product and $\times$ the cross-product. It is the cross-product above that makes the quaternion multiplication noncommutative. The norm is given by

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

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

$$q^* = Sq - Vq = q_r - \imath q_\imath - \jmath q_\jmath - \kappa q_\kappa.$$

Another class of quaternion self-inverse mappings are the quaternion perpendicular involutions, defined as [10]

$$\begin{aligned} q^\imath &= -\imath q \imath = q_r + \imath q_\imath - \jmath q_\jmath - \kappa q_\kappa \\ q^\jmath &= -\jmath q \jmath = q_r - \imath q_\imath + \jmath q_\jmath - \kappa q_\kappa \\ q^\kappa &= -\kappa q \kappa = q_r - \imath q_\imath - \jmath q_\jmath + \kappa q_\kappa \end{aligned} \tag{1}$$

which have the following properties (for any $\eta \in \{\imath, \jmath, \kappa\}$):

$$\begin{aligned} (q^\eta)^\eta &= q, \ (q^\eta)^* = (q^*)^\eta, \\ (q_1 q_2)^\eta &= q_1^\eta q_2^\eta, \ (q_1 + q_2)^\eta = q_1^\eta + q_2^\eta. \end{aligned}$$

Involutions can be observed as a counterpart of the complex conjugate, as they 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

$$\begin{aligned} q_r &= \frac{1}{4}(q + q^\imath + q^\jmath + q^\kappa), \quad q_\imath = \frac{1}{4\imath}(q + q^\imath - q^\jmath - q^\kappa), \\ q_\jmath &= \frac{1}{4\jmath}(q - q^\imath + q^\jmath - q^\kappa), \quad q_\kappa = \frac{1}{4\kappa}(q - q^\imath - q^\jmath + q^\kappa). \end{aligned} \tag{2}$$

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

$$q^* = \frac{1}{2}(q^\imath + q^\jmath + q^\kappa - q) \tag{3}$$

and

$$q = \frac{1}{2}(q^{\imath *} + q^{\jmath *} + q^{\kappa *} - q^*). \tag{4}$$

The quaternion product, norm, conjugate, and involutions will be employed to design quaternion widely linear QESNs and the associated learning algorithms.

III. Augmented Quaternion Statistics and Quaternion Widely Linear Model

This section gives the background necessary for a full exploitation of second-order information of quaternion signals via the quaternion widely linear model, necessary for the design of the proposed QESNs and AQESNs.

A. Augmented Quaternion Statistics

Unlike the real domain where complete second-order statistics of a random vector $\mathbf{q}(k)$ are described by the covariance matrix $\mathbf{R} = E[\mathbf{q}\mathbf{q}^{\mathrm{T}}]$, in the complex and quaternion domains, the covariance matrix is sufficient to describe only second-order circular (proper) signals, which have equal power in data components. For general second-order noncircular (improper) quaternion signals, where powers in the data components may be different, for optimal second-order modeling, we also need to employ complementary covariance matrices (pseudocovariances). These complementary covariance matrices are termed the $\imath$-covariance $\mathbf{P}$, $\jmath$-covariance $\mathbf{S}$ and κ-covariance $\mathbf{T}$, and are given by [16], [29]–[32]

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

Remark 1: Complete second-order characteristics of a quaternion random vector $\mathbf{q}$ are then described by the augmented covariance matrix $\mathbf{R}^a$ of an augmented vector $\mathbf{q}^a = [\mathbf{q}^{\mathrm{T}}, \mathbf{q}^{\imath\mathrm{T}}, \mathbf{q}^{\jmath\mathrm{T}}, \mathbf{q}^{\kappa\mathrm{H}}]^{\mathrm{T}}$, given by

$$\mathbf{R}^a = E[\mathbf{q}^a\mathbf{q}^{a\mathrm{H}}] = \begin{bmatrix} \mathbf{R} & \mathbf{P} & \mathbf{S} & \mathbf{T} \\ \mathbf{P}^{\imath} & \mathbf{R}^{\imath} & \mathbf{T}^{\imath} & \mathbf{S}^{\imath} \\ \mathbf{S}^{\jmath} & \mathbf{T}^{\jmath} & \mathbf{R}^{\jmath} & \mathbf{P}^{\jmath} \\ \mathbf{T}^{\kappa} & \mathbf{S}^{\kappa} & \mathbf{P}^{\kappa} & \mathbf{R}^{\kappa} \end{bmatrix}. \tag{5}$$

Notice that for proper signals, the pseudocovariance matrices $\mathbf{P}$, $\mathbf{S}$, and $\mathbf{T}$ vanish; a signal that obeys this structure has a probability distribution that is rotation invariant with respect to all the six possible pairs of axes [16], [29]–[32]. However, in most of the real-world applications, probability density functions are rotation dependent, and hence require the use of the augmented quaternion statistics.

B. Quaternion Widely Linear Model

To exploit the complete second-order statistics of quaternion-valued signals in linear mean square error (MSE) estimation, we first consider a quaternion-valued MSE estimator given by

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

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

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

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

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

so that using the involutions in (1), we can express each element of a quaternion variable as in (2). This gives, for instance, for 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

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

In other words, to capture the full second-order information available, we should use the original quaternion and its involutions, allowing us to arrive at the widely linear model [16], [29], [30]

$$y = \mathbf{w}^{a\mathrm{T}}\mathbf{q}^a = \mathbf{a}^{\mathrm{T}}\mathbf{q} + \mathbf{b}^{\mathrm{T}}\mathbf{q}^\imath + \mathbf{c}^{\mathrm{T}}\mathbf{q}^\jmath + \mathbf{d}^{\mathrm{T}}\mathbf{q}^\kappa \tag{6}$$

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.

IV. Nonlinear Activation Functions in $\mathbb{H}$

One of the difficulties in the design of hypercomplex RNNs lies in the lack of analytic nonlinear activation functions, as the CRF conditions for analyticity in $\mathbb{H}$ are very stringent [17]. For instance, a CRF differentiable quaternion function $f(q)$ should satisfy

$$\frac{\partial f}{\partial q_r} + \imath\frac{\partial f}{\partial q_\imath} + \jmath\frac{\partial f}{\partial q_\jmath} + \kappa\frac{\partial f}{\partial q_\kappa} = 0 \;\Leftrightarrow\; \frac{\partial f}{\partial q^*} = 0. \tag{7}$$

Only linear quaternion functions and constants fulfill these conditions, yet nonlinear adaptive filtering in $\mathbb{H}$ requires differentiable nonlinear functions. To circumvent the analyticity problem, recent work in [24] adopted the LAC [23], based on a complex-valued representation of a quaternion, to give

$$\frac{\partial f}{\partial q_r} = -\zeta\frac{\partial f}{\partial \alpha} \tag{8}$$

where ζ and α are, respectively, given by

$$\zeta = \frac{\imath q_\imath + \jmath q_\jmath + \kappa q_\kappa}{\alpha}, \quad \alpha = \sqrt{q_\imath^2 + q_\jmath^2 + q_\kappa^2}. \tag{9}$$

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

Proposition 1: The quaternion exponential $e^q = e^{q_r + \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa}$ satisfies the LAC in (8).

Proof: e^q can be expanded using the Euler formula as

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

where ζ and α are defined in (9), to give

$$\frac{\partial e^q}{\partial q_r} = e^q = -\zeta\frac{e^q}{\alpha}. \tag{10}$$

Remark 2: Notice that the quaternion exponential $e^{-q} = e^{-(q_r + \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa)}$ also satisfies the LAC in (8). This is straightforward to show using the same approach as in Proposition 1.

Remark 3: 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 also satisfy the LAC.

For a detailed proof of Remark 3, we refer to [24].

In this paper, we employ a fully quaternion $\tanh(q)$ function to design the QESNs, defined as

$$\tanh(q) = \frac{e^q - e^{-q}}{e^q + e^{-q}} = \frac{e^{2q} - 1}{e^{2q} + 1} \tag{11}$$

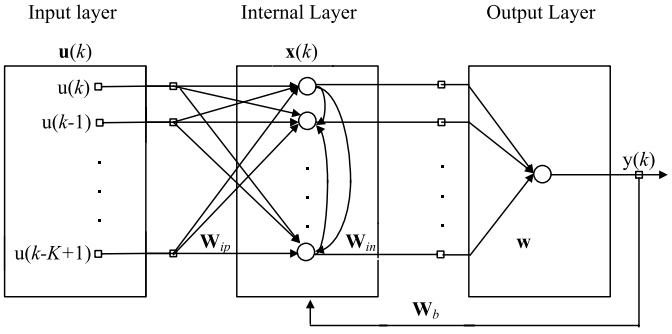

Fig. 1. Architecture of an ESN.

for which first derivative is given by

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

V. Quaternion ESNs

The existence of fully quaternion nonlinear activation functions enables the design of RNNs in $\mathbb{H}$ [24], [25]. In this section, the QESNs and its AQESNs (widely linear) variant are introduced.

A. Standard QESNs

Fig. 1 shows the architecture of an ESN, which is a recurrent discrete-time neural network with K external inputs, N internal neurons (also known as dynamical reservoir), and L readout neurons. The $N \times K$ weight matrix $\mathbf{W}_{\text{ip}}$, the $N \times N$ weight matrix $\mathbf{W}_{\text{in}}$, respectively, contain the connections between the input units and the internal units, models the connections 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 vector $\mathbf{x}(k)$ represents the $N \times 1$ internal state vector, $\mathbf{u}(k)$ the $K \times 1$ input vector, and $\mathbf{y}(k-1)$ is the $L \times 1$ output vector, all at time instant k. The overall network state, denoted by $\mathbf{s}(k)$ is a concatenation of the input $\mathbf{u}(k)$, internal state $\mathbf{x}(k)$ and the delayed output $\mathbf{y}(k)$, and is defined as

$$\mathbf{s}(k) = [u(k), \ldots, u(k-K+1), x_1(k), \ldots, x_N(k), y(k-1), \ldots, y(k-L)]^{\mathrm{T}} \quad (12)$$

while the internal unit dynamics are updated according to

$$\mathbf{x}(k) = \Phi\big(\mathbf{W}_{\text{ip}}\mathbf{u}(k) + \mathbf{W}_{\text{in}}\mathbf{x}(k-1) + \mathbf{W}_b\mathbf{y}(k-1)\big) \quad (13)$$

where $\Phi(\cdot)$ here is a quaternion-valued nonlinear activation of the neurons within the reservoir. The existence of echo state property is critical to ensure adequate operation of the dynamical reservoir of ESNs. This can be achieved by a two-step operation on $\mathbf{W}_{\text{in}}$: 1) randomly choose an internal weight matrix $\mathbf{W}_{\text{in}}$, which is typically drawn from a uniform distribution over a symmetric interval and 2) scale $\mathbf{W}_{\text{in}}$ as $\mathbf{W}_{\text{in}} \leftarrow \mathbf{W}_{\text{in}}/|\lambda_{\max}|$, where $|\lambda_{\max}|$ is the largest absolute eigenvalue of $\mathbf{W}_{\text{in}}$ (spectral radius). The input and feedback connections stored in $\mathbf{W}_{\text{ip}}$ and $\mathbf{W}_b$ can be initialized arbitrarily [26], [28]; recent advances in echo state property analysis can be found in [33]–[35]. The output of a nonlinear output mapping of ESNs is given by [27], [35]–[37]

$$y(k) = \Phi\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big) \quad (14)$$

where $\mathbf{w}(k)$ is the weight vector corresponding to the output layer, and is updated through minimization of a suitable cost functions. The cost function chosen here is the instantaneous squared error, defined as

$$E(k) = \frac{1}{2}|e(k)|^2 = \frac{1}{2}e(k)e^*(k) \quad (15)$$

where $e(k)$ is the instantaneous output error $e(k) = d(k) - y(k)$ and $d(k)$ is the desired (teaching) signal. Note that $E(k)$ is a real-valued function dependent on both quaternion-valued $e(k)$ and $e^*(k)$; the $\mathbb{HR}$ calculus shows that for such functions, the maximum change of gradient lies in the direction of the conjugate gradient, which conforms with the corresponding solutions in $\mathbb{R}$ and $\mathbb{C}$ [38]. Hence, the minimization of $E(k)$ through gradient descent is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu\nabla_{\mathbf{w}^*}E(k) \quad (16)$$

where μ is the step-size, a small positive constant. Using the chain rule, the gradient $\nabla_{\mathbf{w}^*}E(k)$ can be derived as[1]

$$\nabla_{\mathbf{w}^*}E(k) = \frac{1}{2}\frac{\partial|e(k)|^2}{\partial\mathbf{w}^*(k)} = \frac{1}{2}\left(e(k)\frac{\partial e^*(k)}{\partial\mathbf{w}^*(k)} + \frac{\partial e(k)}{\partial\mathbf{w}^*(k)}e^*(k)\right) \quad (17)$$

where

$$e(k) = d(k) - y(k) = d(k) - \Phi\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big), \quad (18)$$

and its conjugate $e^*(k)$, given by[2]

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

The first gradient in (17) can be evaluated as

$$\begin{aligned} \frac{\partial e^*(k)}{\partial\mathbf{w}^*(k)} &= \frac{\partial\big(d^*(k) - \Phi(\mathbf{s}^{\mathrm{H}}(k)\mathbf{w}^*(k))\big)}{\partial\mathbf{w}^*(k)} \\ &= -\frac{\partial\Phi\big(\mathbf{s}^{\mathrm{H}}(k)\mathbf{w}^*(k)\big)}{\partial\mathbf{w}^*(k)} \\ &= -\Phi'\big(\mathbf{s}^{\mathrm{H}}(k)\mathbf{w}^*(k)\big)\frac{\partial\big(\mathbf{s}^{\mathrm{H}}(k)\mathbf{w}^*(k)\big)}{\partial\mathbf{w}^*(k)} \\ &= -\Phi'^*\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)\mathbf{s}^*(k) \end{aligned} \quad (19)$$

where the last step is performed using the $\mathbb{HR}$-derivative introduced in (41) in the Appendix. On the other hand

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

[1]Note that due to the noncommutativity of quaternion product, that is, $q_1q_2 \neq q_2q_1$, the partial derivatives cannot be swapped with $e(k)$ and $e^*(k)$.
[2]This derivation uses the fact that $\Phi^*(q) = \Phi(q^*)$ and $(q_1q_2)^* = q_2^*q_1^*$.

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

$$\frac{\partial\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)}{\partial\mathbf{w}^*(k)} = -\frac{1}{2}\mathbf{s}(k)$$

and hence

$$\frac{\partial e(k)}{\partial\mathbf{w}^*(k)} = \frac{1}{2}\Phi'\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)\mathbf{s}(k).$$

Finally, we arrive at the update of the weight vector $\mathbf{w}(k)$ in the output layer, given by[3]

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) - \mu\nabla_{\mathbf{w}^*}E(k) = \mathbf{w}(k)\\ &\quad+\mu\Big(e(k)\Phi'^*\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)\mathbf{s}^*(k)\\ &\quad-\frac{1}{2}\Phi'\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)\mathbf{s}(k)e^*(k)\Big). \quad (20)\end{aligned}$$

For a linear readout neuron, where $\Phi(q) = q$ and $\Phi'(q) = 1$, the update becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\big(e(k)\mathbf{s}^*(k) - \frac{1}{2}\mathbf{s}(k)e^*(k)\big). \quad (21)$$

B. Augmented QESNs

To make QENSs optimal for the generality of quaternion signals (both second-order circular and noncircular), we use recent developments in augmented quaternion statistics, as described in Section III, to incorporate the widely linear model into the QESNs architecture, which gives the augmented version of QESNs (AQESNs). This means that the network state should be augmented so that

$$\mathbf{s}^a(k) = \big[\mathbf{u}(k), \mathbf{u}^\imath(k), \mathbf{u}^\jmath(k), \mathbf{u}^\kappa(k), \mathbf{x}^a(k), \mathbf{y}(k)\big]^{\mathrm{T}}. \quad (22)$$

This augmented network state design is not limited to the architecture of QESNs but is also applicable for quaternion-valued extensions of other types of RNNs. Since the input weights of QESNs contained in the matrix $\mathbf{W}_{\mathrm{ip}}$ are randomly chosen prior to training, we can use three other matrices $\mathbf{W}_{\mathrm{ip1}}$, $\mathbf{W}_{\mathrm{ip2}}$ and $\mathbf{W}_{\mathrm{ip3}}$ to initialize the weights associated with the input involutions $\mathbf{u}^\imath(k)$, $\mathbf{u}^\jmath(k)$, $\mathbf{u}^\kappa(k)$. In this sense, the internal state dynamics within the AQESNs are updated as

$$\begin{aligned}\mathbf{x}^a(k) = \Phi\big(&\mathbf{W}_{\mathrm{ip}}\mathbf{u}(k) + \mathbf{W}_{\mathrm{ip1}}\mathbf{u}^\imath(k) + \mathbf{W}_{\mathrm{ip2}}\mathbf{u}^\jmath(k)\\ &+ \mathbf{W}_{\mathrm{ip3}}\mathbf{u}^\kappa(k) + \mathbf{W}_{\mathrm{in}}\mathbf{x}(k-1) + \mathbf{W}_b\mathbf{y}(k-1)\big).\end{aligned} \quad (23)$$

Due to the specific properties of the ESN output layer, the output $y(k)$ is now governed by an asymmetric version of the quaternion widely linear model in (6) to yield

$$\begin{aligned}\mathrm{net}(k) &= \mathbf{a}^{\mathrm{T}}(k)\mathbf{v}(k) + \mathbf{b}^{\mathrm{T}}(k)\mathbf{u}^\imath(k) + \mathbf{c}^{\mathrm{T}}(k)\mathbf{u}^\jmath(k) + \mathbf{d}^{\mathrm{T}}(k)\mathbf{u}^\kappa(k)\\ y(k) &= \Phi\big(\mathrm{net}(k)\big)\end{aligned} \quad (24)$$

where $\mathbf{v}(k) = [u(k), \ldots, u(k-K+1), x_1^a(k), \ldots x_N^a(k), y(k-1), \ldots, y(k-L)]^{\mathrm{T}}$ is a subset of the augmented network state $\mathbf{s}^a(k)$ and has the same dimension $(K+N+L)$ as the state vector $\mathbf{s}(k)$ within standard ESNs, however, the internal state dynamics are updated using (23). The weight updates of the output weight vectors $\{\mathbf{a}(k), \mathbf{b}(k), \mathbf{c}(k), \mathbf{d}(k)\}$ are made gradient adaptive according to

$$\begin{aligned}\mathbf{a}(k+1) &= \mathbf{a}(k) - \mu\nabla_{\mathbf{a}^*}E(k)\\ \mathbf{b}(k+1) &= \mathbf{b}(k) - \mu\nabla_{\mathbf{b}^*}E(k)\\ \mathbf{c}(k+1) &= \mathbf{c}(k) - \mu\nabla_{\mathbf{c}^*}E(k)\\ \mathbf{d}(k+1) &= \mathbf{d}(k) - \mu\nabla_{\mathbf{d}^*}E(k).\end{aligned} \quad (25)$$

The gradient $\nabla_{\mathbf{a}^*}E(k)$ in (25) is equivalent to its counterpart $\nabla_{\mathbf{s}^*}E(k)$ in (16) and has the same dimension, hence

$$\mathbf{a}(k+1) = \mathbf{a}(k) + \mu\big(e(k)\Phi'^*(\mathrm{net}(k))\mathbf{v}^*(k) - \frac{1}{2}\Phi'(\mathrm{net}(k))\mathbf{v}(k)\big). \quad (26)$$

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

$$\nabla_{\mathbf{b}^*}E(k) = \frac{1}{2}\left(e(k)\frac{\partial e^*(k)}{\partial\mathbf{b}^*(k)} + \frac{\partial e(k)}{\partial\mathbf{b}^*(k)}e^*(k)\right) \quad (27)$$

where

$$\begin{aligned}\frac{\partial e^*(k)}{\partial\mathbf{b}^*(k)} &= -\frac{\partial\Phi\big(\mathrm{net}^*(k)\big)}{\partial\mathbf{b}^*(k)} = -\Phi'^*\big(\mathrm{net}(k)\big)\frac{\partial\mathrm{net}^*(k)}{\partial\mathbf{b}^*(k)}\\ &= -\Phi'^*\big(\mathrm{net}(k)\big)\mathbf{u}^{\imath*}(k)\end{aligned} \quad (28)$$

and

$$\begin{aligned}\frac{\partial e(k)}{\partial\mathbf{b}^*(k)} &= -\frac{\partial\Phi\big(\mathrm{net}(k)\big)}{\partial\mathbf{b}^*(k)} = -\Phi'\big(\mathrm{net}(k)\big)\frac{\partial\mathrm{net}(k)}{\partial\mathbf{b}^*(k)}\\ &= \frac{1}{2}\Phi'\big(\mathrm{net}(k)\big)\mathbf{u}^\imath(k).\end{aligned} \quad (29)$$

Substituting the partial derivatives in (28) and (29) into the error gradient $\nabla_{\mathbf{b}^*}E(k)$ given in (27) yields

$$\begin{aligned}\mathbf{b}(k+1) = \mathbf{b}(k) + \mu\Big(&e(k)\Phi'^*(\mathrm{net}(k))\mathbf{u}^{\imath*}(k)\\ &-\frac{1}{2}\Phi'(\mathrm{net}(k))\mathbf{u}^\imath(k)e^*(k)\Big).\end{aligned}$$

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

$$\begin{aligned}\mathbf{c}(k+1) = \mathbf{c}(k) + \mu\Big(&e(k)\Phi'^*(\mathrm{net}(k))\mathbf{u}^{\jmath*}(k)\\ &-\frac{1}{2}\Phi'(\mathrm{net}(k))\mathbf{u}^\jmath(k)e^*(k)\Big)\end{aligned}$$

and

$$\begin{aligned}\mathbf{d}(k+1) = \mathbf{d}(k) + \mu\Big(&e(k)\Phi'^*(\mathrm{net}(k))\mathbf{u}^{\kappa*}(k)\\ &-\frac{1}{2}\Phi'(\mathrm{net}(k))\mathbf{u}^\kappa(k)e^*(k)\Big).\end{aligned}$$

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 (22) as

$$\begin{aligned}\mathbf{w}^a(k+1) = \mathbf{w}^a(k) + \mu\Big(&e(k)\Phi'^*(\mathrm{net}(k))\mathbf{s}^{a*}(k),\\ &-\frac{1}{2}\Phi'(\mathrm{net}(k))\mathbf{s}^a(k)e^*(k)\Big)\end{aligned} \quad (30)$$

[3]The factor $1/2$ in the cost function (15) is absorbed into the step-size μ.

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

For a linear readout neuron, where $\Phi(x) = x$ and $\Phi'(x) = 1$, the update becomes

$$\mathbf{w}^a(k+1) = \mathbf{w}^a(k) + \mu\big(e(k)\mathbf{s}^{a*}(k) - \frac{1}{2}\mathbf{s}^a(k)e^*(k)\big). \quad (31)$$

For real-time applications of the proposed QESNs and AQESNs, it is also desirable to use batch mode training where quaternion-valued linear or ridge regression [39] is used to determine the weights $\mathbf{w}$ and $\mathbf{w}^a$ directly over a batch of training data in the MSE sense, which are further applied on the test data. For more detail, we refer to [40].

C. Convergence Analysis of QESNs and AQESNs

To ensure satisfactory performances of QESNs and AQESNs, we employ the convergence criterion, given by [2]

$$E[|\bar{e}(k)|^2] < E[|\tilde{e}(k)|^2] \quad (32)$$

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

$$\begin{aligned}\bar{e}(k) &= d(k) - \Phi\big(\mathbf{w}^{\mathrm{T}}(k+1)\mathbf{s}(k)\big)\\ \tilde{e}(k) &= d(k) - \Phi\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big).\end{aligned} \quad (33)$$

Proposition 2: In the context of QESNs trained by QNGD, the convergence condition in (32) is satisfied for

$$0 < E\Big[5\mu|\Phi'(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k))|^2\|\mathbf{s}(k)\|_2^2\Big] < 1. \quad (34)$$

Proof: The error terms $\bar{e}(k)$ and $\tilde{e}(k)$ in (33) can be related by the first-order Taylor series expansion (TSE) as[4] [41]

$$\begin{aligned}|\bar{e}(k)|^2 &= |\tilde{e}(k)|^2 + \frac{\partial|\tilde{e}(k)|^2}{\partial\mathbf{w}^{\mathrm{T}}(k)}\Delta\mathbf{w}(k) + \frac{\partial|\tilde{e}(k)|^2}{\partial\mathbf{w}^{\imath\mathrm{T}}(k)}\Delta\mathbf{w}^{\imath}(k)\\ &\quad + \frac{\partial|\tilde{e}(k)|^2}{\partial\mathbf{w}^{\jmath\mathrm{T}}(k)}\Delta\mathbf{w}^{\jmath}(k) + \frac{\partial|\tilde{e}(k)|^2}{\partial\mathbf{w}^{\kappa\mathrm{T}}(k)}\Delta\mathbf{w}^{\kappa}(k)\\ &= |\tilde{e}(k)|^2 + 4\Re\left(\frac{\partial|\tilde{e}(k)|^2}{\partial\mathbf{w}^{\mathrm{T}}(k)}\Delta\mathbf{w}(k)\right).\end{aligned} \quad (35)$$

To simplify the derivation of (35), notice that $\Re(q_1q_2) = \Re((q_1q_2)^*) = \Re(q_2^*q_1^*)$ for any pair $\{q_1, q_2\} \in \mathbb{H}$ to give

$$|\tilde{e}(k)|^2 = |\tilde{e}(k)|^2 + 4\Re\left(\Delta\mathbf{w}^{\mathrm{H}}(k)\frac{\partial|\tilde{e}(k)|^2}{\partial\mathbf{w}^*(k)}\right) \quad (36)$$

where the partial derivative $\partial|\tilde{e}(k)|^2/\partial\mathbf{w}^*(k)$ is effectively the gradient of the cost function with respect to $\mathbf{w}^*(k)$, given in (17). The term $\Delta\mathbf{w}^{\mathrm{H}}(k)$ can be expressed using (20) as

$$\begin{aligned}\Delta\mathbf{w}^{\mathrm{H}}(k) &= (\mathbf{w}(k+1) - \mathbf{w}(k))^{\mathrm{H}}\\ &= \mu\bigg(\tilde{e}(k)\Phi'^*\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)\mathbf{s}^*(k)\\ &\qquad - \frac{1}{2}\Phi'\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)\mathbf{s}(k)\tilde{e}^*(k)\bigg)^{\mathrm{H}}\\ &= \mu\bigg(\mathbf{s}^{\mathrm{T}}(k)\Phi'\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)\tilde{e}^*(k)\\ &\qquad - \frac{1}{2}\tilde{e}(k)\mathbf{s}^{\mathrm{H}}(k)\Phi'^*\big(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k)\big)\bigg).\end{aligned} \quad (37)$$

Substitute (17) and (37) into the TSE in (36) to give

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

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

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

The usual statistical independence assumption between $\tilde{e}(k)$ and $\mathbf{s}(k)$ gives

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

Therefore, the convergence condition in (32) is satisfied for

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

Remark 4: The range of the step-size μ for which QESNs converge is given by

$$0 < \mu < \frac{1}{E\big[5|\Phi'(\mathbf{w}^{\mathrm{T}}(k)\mathbf{s}(k))|^2\|\mathbf{s}(k)\|_2^2\big]}. \quad (38)$$

Remark 5: The bound on μ, which ensures the convergence of AQESNs, is given by

$$0 < \mu < \frac{1}{E\big[5|\Phi'(\mathbf{w}^{a\mathrm{T}}(k)\mathbf{s}^a(k))|^2\|\mathbf{s}^a(k)\|_2^2\big]}. \quad (39)$$

Remark 6: For QESNs and AQESNs with a linear readout neuron, the convergence bound can be obtained by replacing the term $\Phi'(\cdot)$ in (38) and (39) with unity.

D. Computational Complexity of QESNs and AQESNs

We next compare the computational complexities of the proposed QESNs and their augmented versions against that of standard real ESNs [36]. At each time instant k, QESNs use (13) to update the internal state dynamics $\mathbf{x}(k)$ and (14), (18), and (20) to train the output layer weights $\mathbf{w}(k)$, while the AQESNs use (23), (24), and (30) to implement the update

[4] In the quaternion domain $\mathbb{H}$, the first-order TSE of a function $f(q) = f(q, q^\imath, q^\jmath, q^\kappa)$ is $df(q) = (\partial f(q)/\partial q)dq + (\partial f(q)/\partial q^\imath)dq^\imath + (\partial f(q)/\partial q^\jmath)dq^\jmath + (\partial f(q)/\partial q^\kappa)dq^\kappa = 4\Re((\partial f/\partial q)dq)$, where $\Re(\cdot)$ is the real part operator [38].

TABLE I
COMPUTATIONAL COMPLEXITIES OF THE ALGORITHMS CONSIDERED

	Multiplications	Additions
Real ESNs [36]	$(\eta N + K + L + 3)N + 2K + 2L + 4$	$(\eta N + K + L + 3)N + 2K + 2L + 3$
QESNs	$(\eta N + K + L + 5)N + 4K + 4L + 4$	$(\eta N + K + L + 4)N + 3K + 3L + 3$
AQESNs	$(\eta N + 4K + L + 5)N + 16K + 4L + 4$	$(\eta N + 4K + L + 4)N + 12K + 3L + 3$

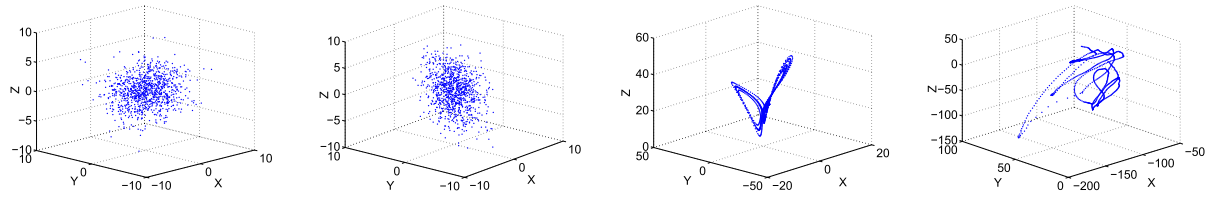

Fig. 2. Geometric view of circularity. For illustration purpose, only the $\imath$-, $\jmath$-, and κ-components of quaternion signals are plotted in the form of a scatter diagram. Observe that only the AR(4) signal driven by circular Gaussian noise in (a) is second-order circular. The AR(4) process driven by noncircular noise, the 3-D Lorenz signal and the real world 3-D Tai Chi body motion signals in (b)–(d) exhibit noncircular distributions.

procedure. The computational complexities of all the considered algorithms are summarized in Table I, where η denotes the degree of connectivity of the internal neurons, calculated as the percentage of the nonzero elements over the size of the internal connection matrix $\mathbf{W}_{\text{in}}$. The complexity of real ESNs was measured in terms of real-valued multiplications and additions, while for the QESNs and AQESNs we counted their quaternion equivalents. The additional computation required by the proposed AQESNs over standard QESNs results from the incorporation of the widely linear model (input augmentation), that is $[\mathbf{u}(k), \mathbf{u}^{\imath}(k), \mathbf{u}^{\jmath}(k), \mathbf{u}^{\kappa}(k)]$ instead of $\mathbf{u}(k)$ itself, into the input layer of QESNs, as illustrated in (22). However, their computational complexities are still comparable, as the internal layer takes the bulk of calculations, that is, ηNN in both multiplications and additions. This is more pronounced for large scale QESNs, where $N \gg K$ and $N \gg L$.

E. Advantages of AQESNs Over Existing RNNs in $\mathbb{H}$

The proposed AQESNs exhibit the following novelties and advantages over the existing quaternion-valued schemes.

1) The quaternion multilayer perceptron [42] and the split quaternion nonlinear adaptive filtering algorithms [20] use a split quaternion function that treats each quaternion component separately (as a real data channel) passed through a real smooth nonlinearity. Hence, the noncommutativity aspect of the quaternion algebra is neglected, and such schemes do not exploit the full potential of the processing in the quaternion domain. The proposed AQESNs employ full quaternion nonlinearity in $\mathbb{H}$ instead of split quaternion function $\in \mathbb{R}$, and preserve the cross-information within the data components.
2) Compared with the quaternion Hopfield neural networks [19], [43] and fully connected RNNs (FCRNNs) [25] using the fully quaternion nonlinearity, the proposed AQESNs employ a randomly and sparsely connected dynamical reservoir, require training only for the weights connecting the hidden layer and the readout neurons, hence significantly reducing computational complexity; this advantage is more pronounced in designing large-scale quaternion RNNs.
3) Compared with the quaternion nonlinear adaptive filters [24] based on a simple feedforward FIR structure, the proposed AQESNs employ the widely linear model in the context of general quaternion RNNs with three layers and feedback, and thus have the ability to better capture the available noncircular statistics.

VI. SIMULATIONS

We performed simulations in the one step ahead prediction setting, for both benchmark synthetic proper and improper 3-D and 4-D signals and for real-world improper 3-D body motion data. For a fair performance assessment, the length of the training sequence was set to 4000 for all the considered signals. Two input neurons and two output neurons were used and we followed the rule of thumb that the number of internal neurons N within the dynamical reservoir should be about one tenth of the data length [27], [40]. To illustrate the robustness of the proposed QESNs, different dynamical reservoir sizes in the order of 100 were investigated in the simulations. The randomly selected input, internal, and feedback weights $\mathbf{W}_{\text{ip}}$, $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_b$ were generated from a uniform distribution in the range $[-1, 1]$, and the spectral radius $\rho(\mathbf{W}_{\text{in}})$ was scaled to $\rho = 0.8$. The internal weight connections contained in $\mathbf{W}_{\text{in}}$ were sparse, with the interconnectivity ratio $\eta = 5\%$. The step-size μ of the gradient descent algorithms used to train the output layers of QESNs and AQESNs was set at $\mu = 0.01$. The quantitative performance measure was the prediction gain R_p, defined as

$$\text{R}_p = 10\log_{10}\frac{\hat{\sigma}_q^2}{\hat{\sigma}_e^2} \tag{40}$$

TABLE II

COMPARISON OF PREDICTION GAINS R_p OF STANDARD QESNS AND AQESNS WITH 200 NEURONS IN THE DYNAMICAL RESERVOIR FOR THE VARIOUS CLASSES OF SIGNALS CONSIDERED. THE RESULTS WERE OBTAINED BY AVERAGING 100 INDEPENDENT SIMULATION TRIALS

R_p[dB]	Circular AR(4)	Noncircular AR(4)	3D Lorenz	3D Tai Chi body motion
QESNs (linear output layer)	3.57	2.58	17.73	9.53
AQESNs (linear output layer)	3.51	3.46	18.92	17.24
QESNs (nonlinear output layer)	3.60	2.64	18.03	9.86
AQESNs (nonlinear output layer)	3.53	3.57	19.28	17.64

TABLE III

PERCENTAGE OF AQESNS OUTPERFORMING STANDARD QESNS FOR NONCIRCULAR SIGNALS OVER 100 INDEPENDENT INITIALIZATIONS, $N = 200$ NEURONS WERE EMPLOYED IN THE DYNAMICAL RESERVOIR

Noncircular AR(4)	3D Lorenz	3D Tai Chi body motion
100%	98%	97%

where $\hat{\sigma}_q^2$ and $\hat{\sigma}_e^2$ denote the estimated variances of the input and the prediction error. The test signals employed were:

1) a stable circular linear autoregressive AR(4) process, given by [4]

$$q(k) = 1.79q(k-1) - 1.85q(k-2) + 1.27q(k-3) - 0.41q(k-4) + n(k)$$

driven by circular quaternion white Gaussian noise

$$n(k) = n_r(k) + \imath n_\imath(k) + \jmath n_\jmath(k) + \kappa n_\kappa(k)$$

where $n_r(k)$, $n_\imath(k)$, $n_\jmath(k)$ and $n_\kappa(k)$ are independent realizations of real-valued WGN $\sim \mathcal{N}(0,1)$;

2) the same AR(4) process driven by quaternion noncircular noise, where $n_r = \mathcal{N}(0,1)$, $n_\imath = -0.6n_r + \mathcal{N}(0,1)$, $n_\jmath = 0.8n_\imath + \mathcal{N}(0,1)$, $n_\kappa = 0.8n_r - 0.4n_\imath + \mathcal{N}(0,1)$ [24];

3) the noncircular chaotic Lorenz signal, governed by coupled partial differential equations [44]

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

where $\alpha = 10$, $\rho = 28$, and $\beta = 8/3$;

4) a real-world 3-D noncircular and nonstationary body motion signal. 3-D motion data were recorded using the XSense MTx 3DOF orientation tracker, placed on the left and the right arms of an athlete performing Tai Chi movements. The movement of the left arm was used as a pure quaternion input.

Fig. 2 shows the 3-D scatter plots of the quaternion-valued signals considered, providing a geometric view of noncircularity. Observe that only the AR(4) process in Fig. 2(a) had a rotation invariant distribution (circular), while the other signals considered were noncircular.

Table II compares the averaged prediction gains R_p [dB] for standard QESNs and AQESNs with both linear and nonlinear readout neurons. For circular AR(4) signals, the performances of standard QESNs and AQESNs were comparable, since for second-order circular data the widely linear model simplifies into the strictly linear one, as the weights associated to the involutions of the input vector, that is $\{\mathbf{b}, \mathbf{c}, \mathbf{d}\}$ in (24), vanish. For the noncircular signals considered, there was a significant

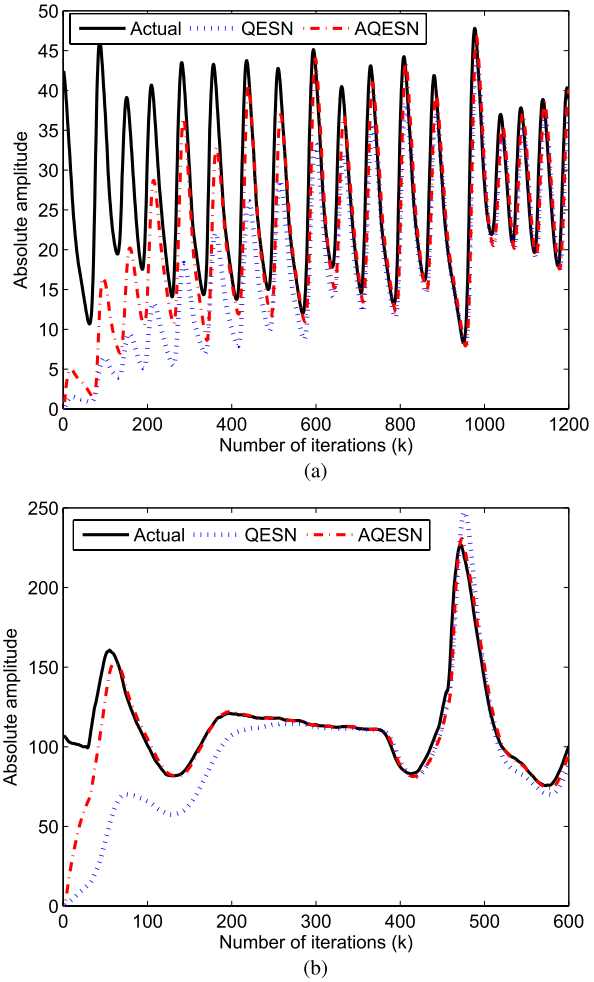

Fig. 3. Performance of QESNs and AQESNs on a one step ahead prediction of single-trial noncircular Lorenz signal and 3-D Tai Chi body motion data. Absolute values of the actual and predicted signals are plotted. (a) 3-D Lorenz signal. (b) 3-D Tai Chi body motion signal.

improvement in the prediction gain when the AQESNs were employed. In all cases, the advantage of employing nonlinear readout neurons within ESNs was justified. The enhanced performance of AQESNs over standard QESNs for noncircular signals is quantified in Table III. Fig. 3(a) and (b) shows the overall prediction performance over the three dimensions of the noncircular 3-D Lorenz and Tai Chi data for QESNs with 200 internal neurons and nonlinear readout neurons. In both cases, the AQESNs tracked the actual signal more accurately than the standard QESNs. Due to the random natures of $\{\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{ip}}, \mathbf{W}_b\}$, the AQESNs cannot guarantee improved performance over its standard version for every trial, due to the network initialization issues. However, on the

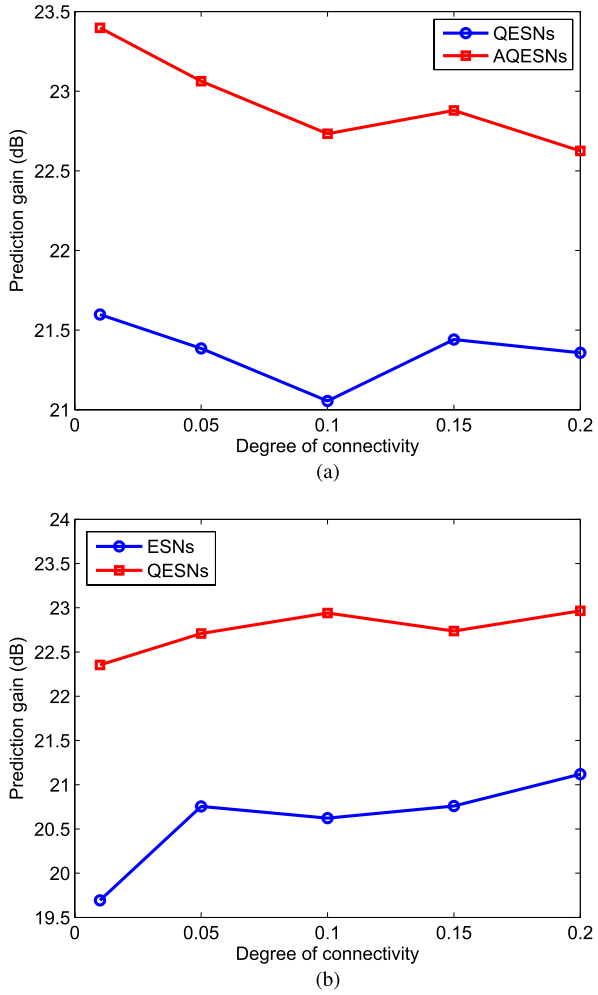

Fig. 4. Performance comparison of standard and AQESNs with 500 neurons in the dynamical reservoir, and different degrees of connectivity. One-step ahead prediction was performed for the noncircular 3-D chaotic Lorenz and 3-D Tai Chi body motion signals. The results were obtained by averaging 100 independent initializations. (a) 3-D Lorenz signal. (b) 3-D Tai Chi body motion signal.

average, as shown in Table III, the AQESNs outperformed the corresponding standard ones in over 98% of the trials.

In the design of ESNs, a key system requirement is a rich variety of dynamics of different internal units. In practice, this is achieved by generating the internal weight matrix $\mathbf{W}_{\text{in}}$ with sparse connections so that the reservoir contains many loosely coupled subsystems [26], [40]. However, an optimal degree of connectivity may vary for signals with different dynamics. Fig. 4(a) and (b) shows the performances of standard and AQESNs with nonlinear readout neurons and 500 neurons in the dynamical reservoir over the degrees of connectivity $\{1\%, 5\%, 10\%, 15\%, 20\%\}$. For the noncircular chaotic 3-D Lorenz signal, a small degree of connectivity at 1% gave the best performance for both QESNs and AQESNs, while a 20% degree of connectivity favored both algorithms for noncircular 3-D Tai Chi body motion data. The widely linear QESNs accounted for general noncircular signals and showed performance advantage over standard QESNs in all cases.

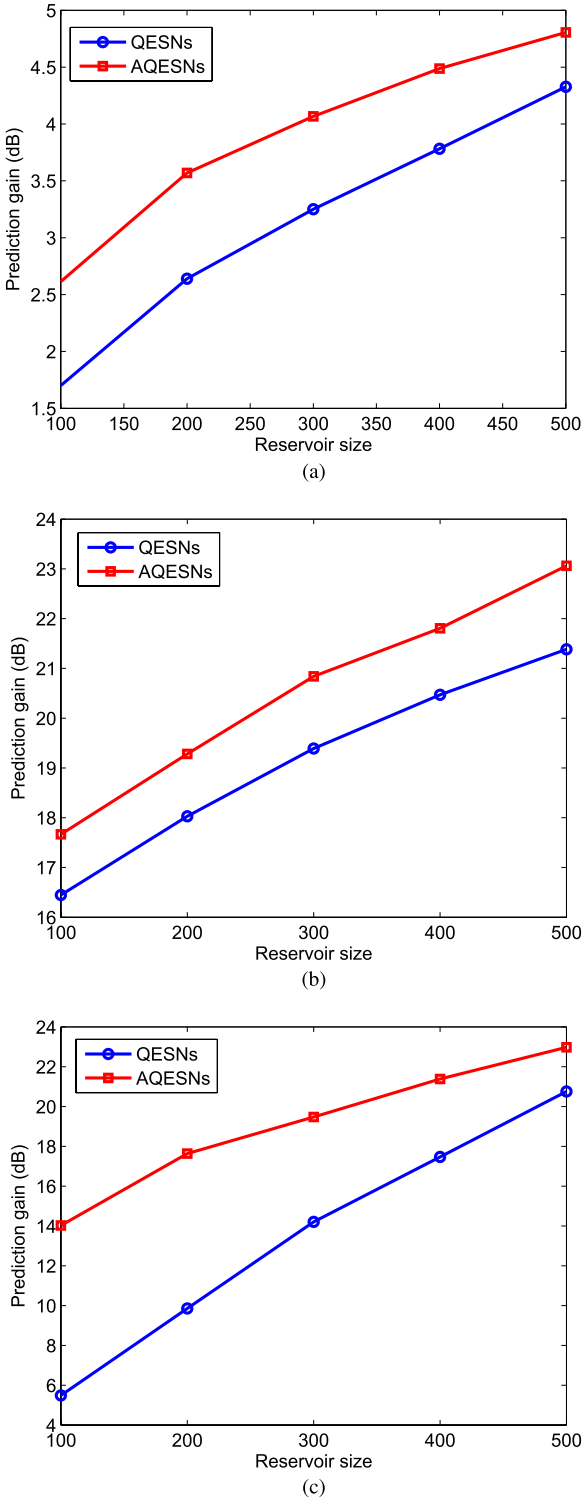

Fig. 5. Performance comparison of standard and AQESNs with the degree of connectivity $\eta = 5\%$, and different dynamical reservoir sizes. One-step ahead prediction was performed for the noncircular 3-D chaotic Lorenz and 3-D Tai Chi body motion signals. The results were obtained by averaging 100 independent initializations. (a) Noncircular AR(4) signal. (b) 3-D Lorenz signal. (c) 3-D Tai Chi body motion signal.

To further illustrate the advantage of using augmented quaternion statistics within QESNs, we compared the performances of both augmented and standard QESNs against the size of dynamical reservoir, an important parameter that

influences the performance of ESNs, as it reflects their universal approximation ability. Generally speaking, an ESN with a larger reservoir can learn the signal dynamics with higher accuracy [28]. This is confirmed in Fig. 5(a)–(c), where the performances of QESNs and AQESNs were investigated against the reservoir size. In all cases, the AQESNs outperformed their standard counterparts.

VII. Conclusion

Fully QESNs have been proposed for the processing of hypercomplex (3-D and 4-D) signals. This has been achieved by making use of recently introduced locally analytical quaternion-valued activation functions, allowing for gradient decent-based training of QESNs. To 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 advantage of the proposed AQESNs over standard QESNs has been illustrated by simulations over a range of noncircular synthetic signals and for real-world noncircular 3-D body motion recordings.

Appendix

$\mathbb{HR}$-Calculus and Quaternion Gradient Operations

The $\mathbb{HR}$-calculus enables the differentiation of both analytic and nonanalytic functions of quaternion variables [38]. It makes possible to circumvent the stringent CRF conditions, which are satisfied only by linear functions and constants [17]. The $\mathbb{HR}$ and $\mathbb{HR}^*$-derivatives within the $\mathbb{HR}$-calculus are given, respectively, by [38]

$$\frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q} = \frac{1}{4}\left(\frac{\partial f}{\partial q_r} - \imath\frac{\partial f}{\partial q_\imath} - \jmath\frac{\partial f}{\partial q_\jmath} - \kappa\frac{\partial f}{\partial q_\kappa}\right) \tag{41}$$

and

$$\frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^*} = \frac{1}{4}\left(\frac{\partial f}{\partial q_r} + \imath\frac{\partial f}{\partial q_\imath} + \jmath\frac{\partial f}{\partial q_\jmath} + \kappa\frac{\partial f}{\partial q_\kappa}\right). \tag{42}$$

For example, to use the $\mathbb{HR}$-derivative in (41) on an analytic function $f(q) = q$, we first need to express it in terms of the involutions $\{q, q^\imath, q^\jmath, q^\kappa\}$ using (3), 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 $f'(q) = 1$. However, in quaternion-valued statistical signal processing, a common optimization objective is to minimize a positive real-valued cost function of quaternion variables, typically $f(q, q^*) = qq^*$. Such 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 (41) and (42). 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 $\partial f(q,q^*)/\partial q^* = \partial(qq^*)/\partial q^* = \partial q/\partial q^* \cdot q^* + q \cdot \partial q^*/\partial q^* = -q^*/2 + q$. Note that the main difference between the $\mathbb{HR}$ calculus and the corresponding $\mathbb{CR}$ calculus [4], [45], [46] in the complex domain lies in the derivative $\partial z/\partial z^* = 0$ for $z = z_r + \imath z_\imath \in \mathbb{C}$, whereas $\partial q/\partial q^* = -1/2$ for $q \in \mathbb{H}$; this is due to the fact that in the quaternion domain $\mathbb{H}$, q and its conjugate q^* are related in the way defined in (3) and (4), whereas in $\mathbb{C}$ such relationship does not exist.

References

[1] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals, Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

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

[3] A. Hirose, *Complex-Valued Neural Networks: Theories and Applications*. Singapore: World Scientific, 2003.

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

[5] I. N. Aizenberg, N. N. Aizenberg, and J. P. L. Vandewalle, *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Norwell, MA, USA: Kluwer, 2000.

[6] S. Jankowski, A. Lozowski, and J. M. Zurada, "Complex-valued multistate neural associative memory," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1491–1496, Sep. 1996.

[7] T. Nitta, *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*. New York, NY, USA: Information Science Reference, 2009.

[8] S. B. Choe and J. J. Faraway, "Modeling head and hand orientation during motion using quaternions," *J. Aerosp.*, vol. 113, no. 1, pp. 186–192, 2004.

[9] C. F. Karney, "Quaternions in molecular modeling," *J. Molecular Graph. Model.*, vol. 25, no. 5, pp. 595–604, 2007.

[10] T. A. Ell and S. J. Sangwine, "Hypercomplex Fourier transforms of color images," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 22–35, Jan. 2007.

[11] N. L. Bihan and J. Mars, "Singular value decomposition of quaternion matrices: A new tool for vector-sensor signal processing," *Signal Process.*, vol. 84, no. 7, pp. 1177–1199, 2004.

[12] N. L. Bihan, S. Miron, and J. I. Mars, "MUSIC algorithm for vector-sensors array using biquaternions," *IEEE Trans. Signal Process.*, vol. 55, no. 9, pp. 4523–4533, Sep. 2007.

[13] C. C. Took, G. Strbac, K. Aihara, and D. P. Mandic, "Quaternion-valued short-term joint forecasting of three-dimensional wind and atmospheric parameters," *Renew. Energy*, vol. 36, no. 6, pp. 1754–1760, 2011.

[14] D. Choukkroun, I. Y. Bar-Itzhack, and Y. Ohsman, "Novel quaternion Kalman filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 174–190, Jan. 2006.

[15] C. Jahanchahi and D. P. Mandic, "A class of quaternion Kalman filters," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 533–544, Mar. 2014.

[16] C. C. Took and D. P. Mandic, "A quaternion widely linear adaptive filter," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4427–4431, Aug. 2010.

[17] A. Sudbery, "Quaternionic analysis," *Math. Proc. Cambridge Philosoph. Soc.*, vol. 85, no. 2, pp. 199–225, 1979.

[18] T. Nitta, "A backpropagation algorithm for neural networks based an 3D vector product," in *Proc. IJCNN*, vol. 1. Nagoya, Japan, Oct. 1993, pp. 589–592.

[19] T. Isokawa, H. Nishuimura, N. Kamiura, and N. Matsui, "Associative memory in quaternionic Hopfield neural network," *Int. J. Neural Syst.*, vol. 18, no. 2, pp. 135–145, 2008.

[20] B. C. Ujang, C. C. Took, and D. P. Mandic, "Split quaternion nonlinear adaptive filtering," *Neural Netw.*, vol. 23, no. 3, pp. 426–434, 2010.

[21] R. J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity," in *Back-Propagation: Theory, Architecture, and Applications*, Y. Chauvin and D. E. Rumelhart, Eds. Hillsdale, NJ, USA: Lawrence Erlbaum, 1995, ch. 13, pp. 433–486.

[22] S. L. Goh and D. P. Mandic, "An augmented CRTRL for complex-valued recurrent neural networks," *Neural Netw.*, vol. 20, no. 10, pp. 1061–1066, 2007.

[23] S. De Leo and P. P. Rotelli, "Quaternionic analyticity," *Appl. Math. Lett.*, vol. 16, no. 7, pp. 1077–1081, 2003.

[24] B. C. Ujang, C. C. Took, and D. P. Mandic, "Quaternion-valued nonlinear adaptive filtering," *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1193–1206, Aug. 2011.

[25] B. C. Ujang, C. C. Took, and D. P. Mandic, "On quaternion analyticity: Enabling quaternion-valued nonlinear adaptive filtering," in *Proc. IEEE ICASSP*, Kyoto, Japan, Mar. 2012, pp. 2117–2120.
[26] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.
[27] H. Jaeger, M. Likoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Netw.*, vol. 20, no. 3, pp. 335–352, 2007.
[28] M. LukošEvičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, 2009.
[29] J. Via, D. Ramirez, and I. Santamaria, "Properness and widely linear processing of quaternion random vectors," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3502–3515, Jul. 2010.
[30] C. Cheong-Took and D. P. Mandic, "Augmented second-order statistics of quaternion random signals," *Signal Process.*, vol. 91, no. 2, pp. 214–224, 2011.
[31] J. Navarro-Moreno, R. M. Fernandez-Alcala, and J. C. Ruiz-Molina, "A quaternion widely linear series expansion and its applications," *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 868–871, Dec. 2012.
[32] J. Navarro-Moreno, J. C. Ruiz-Molina, A. Oya, and J. M. Quesada-Rubio, "Detection of continuous-time quaternion signals in additive noise," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, pp. 1–7, 2012.
[33] M. Buehner and P. Young, "A tighter bound for the echo state property," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 820–824, May 2006.
[34] M. C. Ozturk, D. Xu, and J. C. Príncipe, "Analysis and design of echo state networks," *Neural Comput.*, vol. 19, no. 1, pp. 111–138, 2007.
[35] B. Zhang, D. J. Miller, and Y. Wang, "Nonlinear system modeling with random matrices: Echo state networks revisited," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 175–182, Jan. 2012.
[36] Y. Xia, B. Jelfs, M. M. Van Hulle, J. C. Principe, and D. P. Mandic, "An augmented echo state network for nonlinear adaptive filtering of complex noncircular signals," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 74–83, Jan. 2011.
[37] S. P. Chatzis and Y. Demiris, "Echo state Gaussian process," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1435–1445, Sep. 2011.
[38] D. P. Mandic, C. Jahanchahi, and C. C. Took, "A quaternion gradient operator and its applications," *IEEE Signal Process. Lett.*, vol. 18, no. 1, pp. 47–50, Jan. 2011.
[39] F. A. Tobar, S.-Y. Kung, and D. P. Mandic, "Multikernel least mean square algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 265–277, Feb. 2014.
[40] H. Jaeger, "The echo state approach to analyzing and training neural networks," German Nat. Res. Inst. Inform. Technol., Sankt Augustin, Germany, Tech. Rep. 148, 2002.
[41] E. Soria-Olivas, J. Calpe-Maravilla, J. F. Guerrero-Martinez, M. Martinez-Sober, and J. Espi-Lopez, "An easy demonstration of the optimum value of the adaptation constant in the LMS algorithm [FIR filter theory]," *IEEE Trans. Educ.*, vol. 41, no. 1, p. 81, Feb. 1998.
[42] P. Arena, L. Fortuna, G. Muscato, and M. G. Xibilia, "Multilayer perceptrons to approximate quaternion valued functions," *Neural Netw.*, vol. 10, no. 2, pp. 335–342, 1997.
[43] M. Yoshida, Y. Kuroe, and T. Mori, "Models of Hopfield-type quaternion neural networks and their energy functions," *Int. J. Neural Syst.*, vol. 15, nos. 1–2, pp. 129–135, 2005.
[44] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering (Studies in Nonlinearity)*. Boulder, CO, USA: Westview, 2001.
[45] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *Proc. Inst. Elect. Eng. F, Commun., Radar, Signal Process.*, vol. 130, no. 1, pp. 11–16, Feb. 1983.
[46] K. Kreutz-Delgado, "The complex gradient operator and the $\mathbb{CR}$-calculus," Dept. Elect. Comput. Eng., Univ. California, San Diego, CA, USA, Tech. Rep. ECE275A, 2006.

Yili Xia (M'11) received the B.Eng. degree in information engineering from Southeast University, Nanjing, China, in 2006, the M.Sc. (Hons.) degree in communications and signal processing from the Department of Electrical and Electronic Engineering, Imperial College London, London, U.K., in 2007, and the Ph.D. degree in adaptive signal processing from Imperial College London in 2011.

He has been a Research Associate with Imperial College London since the graduation, and is currently an Associate Professor with the School of Information and Engineering, Southeast University. His current research interests include linear and nonlinear adaptive filters, and complex valued and quaternion valued statistical analysis.

Cyrus Jahanchahi received the M.Eng. degree in electrical and electronic engineering from Imperial College London, London, U.K., where he is currently pursuing the Ph.D. degree in signal processing.

His current research interests include linear and nonlinear adaptive filters, and quaternion valued statistical analysis.

Danilo P. Mandic (M'99–SM'03–F'12) is a Professor of Signal Processing with Imperial College London, London, U.K., where he has been involved in nonlinear adaptive signal processing and nonlinear dynamics. He has been a Guest Professor with Katholieke Universiteit Leuven, Leuven, Belgium, the Tokyo University of Agriculture and Technology, Tokyo, Japan, and Westminster University, London, U.K., and a Frontier Researcher with RIKEN, Wako, Japan. He has two research monographs titled *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability* (West Sussex, U.K.: Wiley, 2001) and *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models* (West Sussex, U.K.: Wiley, 2009), an edited book titled *Signal Processing Techniques for Knowledge Extraction and Information Fusion* (New York, NY, USA: Springer, 2008), and more than 200 publications on signal and image processing.

Prof. Mandic has been a member of the IEEE Technical Committee on Signal Processing Theory and Methods, and an Associate Editor of the IEEE SIGNAL PROCESSING MAGAZINE, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON NEURAL NETWORKS. He has produced award winning papers and products resulting from his collaboration with the industry.