ON QUATERNION ANALYTICITY: ENABLING QUATERNION-VALUED NONLINEAR ADAPTIVE FILTERING

Bukhari Che Ujang, Clive Cheong Took and Danilo P. Mandic

Electrical and Electronic Engineering Department, Imperial College London, SW7 2BT, UK
E-mails: {che.che-ujang07, c.cheong-took, d.mandic}@ic.ac.uk

ABSTRACT

The strict Cauchy-Riemann-Fueter (CRF) analyticity conditions establish that only linear quaternion-valued functions are analytic, prohibiting the development of quaternion-valued nonlinear adaptive filters for the recurrent neural network architecture (RNN). In this work, the requirement of local analyticity in gradient based learning is exercised and proposes to use the local analyticity condition (LAC) to introduce quaternion-valued nonlinear feedback adaptive filters. The introduced class of algorithms make full use of quaternion algebra and provide generic extensions of the corresponding real and complex solutions. Simulations in the prediction setting support the analysis presented.

Index Terms— Nonlinear Adaptive Filtering, Recurrent Neural Networks, Quaternion Analyticity, IIR filters, RTRL

1. INTRODUCTION

The emergence of vector sensors in many areas of technology, robotics and biomedicine has highlighted the need to develop fundamental algorithmic solutions for the class of three- and four-dimensional signals. This class of signals is typically encountered in 3D body motion sensors and aeronautic applications, which require precise descriptors of 3D rotations and orientations. In this respect, quaternions do not suffer from singularities inherent to the vector algebra in $\mathbb{R}^3$ and $\mathbb{R}^4$, such as the gimbal lock, and therefore are widely used in the representation of three- and four-dimensional measurements. Enabling tools for quaternion signal processing include spectrum analysis [1], algebraic matrix decompositions [2], and augmented statistics [3]. For practical filtering and tracking applications, current quaternion-valued adaptive filters have addressed the linear system modelling of both the finite impulse response (FIR) [4] and the infinite impulse response (IIR) [5], and more recently, the more generic class of nonlinear systems and signals [6].

Quaternion-valued nonlinear filtering algorithms makes use of elementary transcendental functions (ETF) [7], which do not satisfy the Cauchy-Riemann-Fueter (CRF) conditions; in fact these strict conditions are met by only linear quaternion-valued functions. To circumvent this analyticity problem, we adopted an approximate alternative to the CRF conditions, that is, the local analyticity condition (LAC) [8]. It treats the quaternion variable similarly to a complex variable and can only guarantee the first-order differentiability of single variable quaternion functions at a point. Notice, however, that for most gradient based learning algorithms the first order derivative is only adequate, enabling the derivation of nonlinear algorithms as shown in [6].

This class of nonlinear algorithms in [6] was based on the feedforward architecture and requires a long filter length for the modelling of systems with long term correlations. For such a case, the IIR architecture is more appropriate due to the feedback, as these can model long term correlations with a small-scale model. For completeness, our aim is to investigate the suitability of LAC in the derivation of gradient-based learning algorithms for feedback architectures and to provide building block for recurrent neural networks in the context of quaternion-valued signal processing. The LAC will then allow us to make use of the 'fully' rather than the 'split' quaternion functions. Similarly to the complex domain [7, 9], these 'fully' quaternion functions permit rigorous treatment of the cross-information across the data channels, in contrast to the componentwise operation of the 'split' quaternion functions. The use of the recurrent architecture and the fully quaternion functions will thus enhance the generality of the existing class of quaternion-valued adaptive filtering algorithms [4–6]. To demonstrate the usefulness of our proposed technique, we shall assess the performance of the proposed Quaternion-valued RTRL (QRTRL) against existing techniques for the prediction of the synthetic 3D Lorenz attractor and real-world 3D body motion signals.

2. ELEMENTS OF QUATERNION ALGEBRA

A quaternion variable q can be expressed as

$$q = [q_a, \bar{q}] = q_a + q_b\imath + q_c\jmath + q_d\kappa \tag{1}$$

where $q_a, q_b, q_c, q_d \in \mathbb{R}$ and $\imath$, $\jmath$, κ are both orthogonal unit vectors and imaginary units variable. q is called a pure quaternion when its real part vanishes, i.e., $q_a = 0$. The relationships of these orthogonal unit vectors are given as

$$\begin{aligned} &\imath\jmath = -\jmath\imath = \kappa; \quad \jmath\kappa = -\kappa\jmath = \imath; \quad \kappa\imath = -\imath\kappa = \jmath; \\ &\imath\jmath\kappa = \imath^2 = \jmath^2 = \kappa^2 = -1 \end{aligned} \tag{2}$$

Quaternion multiplication is calculated as

$$wx = [w_a, \bar{w}][x_a, \bar{x}] = [w_a x_a - \bar{w}\cdot\bar{x}, w_a\bar{x} + x_a\bar{w} + \bar{w}\times\bar{x}] \tag{3}$$

where the symbols "$\cdot$" and "$\times$" denote respectively the dot-product and cross-product. Observe that the quaternion multiplication is non-commutative because of the outer product between $\bar{w}$ and $\bar{x}$.

Similar to the complex case, the quaternion conjugate and the norm square are respectively given as $q^* = [q_a, \bar{q}]^* = [q_a, -\bar{q}]$ and $\| q \|_2^2 = qq^* = q^*q$. From this point onwards, all values are quaternion-valued unless mentioned otherwise.

3. QUATERNION-VALUED NONLINEAR FUNCTIONS

The Cauchy-Riemann (CR) equations which govern the analyticity in the complex domain $\mathbb{C}$ are given by

$$\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\imath = 0 \Leftrightarrow \frac{\partial f}{\partial z^*} = 0 \tag{4}$$

where $f(z) = u(x,y) + v(x,y)\imath$ and $z = x + y\imath$. Analyticity in the quaternion domain $\mathbb{H}$ is stricter than that in $\mathbb{C}$ and is defined by the Cauchy-Riemann-Fueter (CRF) conditions given by

$$\frac{\partial f}{\partial q_a} + \frac{\partial f}{\partial q_b}\imath + \frac{\partial f}{\partial q_c}\jmath + \frac{\partial f}{\partial q_d}\kappa = 0 \tag{5}$$

It is important to note that only linear quaternion functions satisfy the CRF conditions prohibiting nonlinear adaptive filtering in $\mathbb{H}$ [6].

To relax the restriction imposed by the CRF conditions, the "local" analyticity condition (LAC) was used and is defined as [8]

$$\frac{\partial f}{\partial q_a} = -\frac{\partial f}{\partial \alpha}\hat{\zeta} \tag{6}$$

where $\hat{\zeta}$ and α are given by

$$\hat{\zeta} = \frac{q_b\imath + q_c\jmath + q_d\kappa}{\alpha}; \qquad \alpha = \sqrt{q_b^2 + q_c^2 + q_d^2} \tag{7}$$

The term "local" reflects the fact that the "imaginary" unit $\hat{\zeta}$ is dependent on the values of q_b, q_c and q_d.

The class of functions that satisfy the LAC in (6) are shown to be fully quaternion functions and their descriptions are given in [6]. The locally analytic exponential function which is used as a building block for fully quaternion functions, can be expressed

$$\mathrm{e}^q = \mathrm{e}^{q_a + q_b\imath + q_c\jmath + q_d\kappa} = \mathrm{e}^{q_a}\mathrm{e}^{q_b\imath + q_c\jmath + q_d\kappa} \tag{8}$$

Then, the definition for the locally analytic $\tanh(q)$ and its derivative are given by [6]

$$\tanh(q) = \frac{\mathrm{e}^{2q} - 1}{\mathrm{e}^{2q} + 1}; \qquad \frac{\partial \tanh(q)}{\partial q} = \mathrm{sech}^2(q) \tag{9}$$

The entire class of fully quaternion functions and their derivatives are elaborated in [6].

4. ANALYSIS OF QUATERNION-VALUED FUNCTIONS

We next show that the fully-quaternion functions are superior in capturing the cross-correlations between the dimensions compared to the split-quaternion functions. Consider again the function e^q as it serves as a building block to construct other quaternion elementary transcendental functions, that is

$$\mathrm{e}^q = \mathrm{e}^{q_a} + \mathrm{e}^{q_b}\imath + \mathrm{e}^{q_c}\jmath + \mathrm{e}^{q_d}\kappa = y_a + y_b\imath + y_c\jmath + y_d\kappa \tag{10}$$

where y_a, y_b, y_c and y_d are the real-valued elements of the quaternion-valued output. Analyzing the output component y_a shows that $E\{y_a\} = E\{\mathrm{e}^{q_a}\}$ contains the corresponding input q_a thus not accounting for the interchannel couplings.

Now consider a "fully" e^q function that gives the output

$$\begin{aligned}
\mathrm{e}^q &= \mathrm{e}^{q_a}\mathrm{e}^{q_b\imath + q_c\jmath + q_d\kappa} \\
&= \mathrm{e}^{q_a}\Bigg(\cos(\sqrt{q_b^2 + q_c^2 + q_d^2}) + \frac{q_b \sin(\sqrt{q_b^2 + q_c^2 + q_d^2})}{\sqrt{q_b^2 + q_c^2 + q_d^2}}\imath \\
&+ \frac{q_c \sin(\sqrt{q_b^2 + q_c^2 + q_d^2})}{\sqrt{q_b^2 + q_c^2 + q_d^2}}\jmath + \frac{q_d \sin(\sqrt{q_b^2 + q_c^2 + q_d^2})}{\sqrt{q_b^2 + q_c^2 + q_d^2}}\kappa\Bigg) \\
&= y_a + y_b\imath + y_c\jmath + y_d\kappa
\end{aligned} \tag{11}$$

Examining the output y_a componentwise shows that $E\{y_a\} = E\Big\{\mathrm{e}^{q_a}\cos(\sqrt{q_b^2 + q_c^2 + q_d^2})\Big\}$ represents a nonlinear combination of all the input components q_a, q_b, q_c, q_d, and therefore accounts for the internal couplings. This analysis holds true for the other components of a quaternion as well.

5. FCRNN ALGORITHMS IN $\mathbb{H}$

The Fully Connected Recurrent Neural Network (FCRNN) consists of N neurons and p external inputs. The network has two distinct layers consisting a feedback layer and a layer of processing elements. In order to make these terms consistent with past recurrent neural network literature, we let $y_l(k)$ denote the quaternion-valued output of each neuron, $l = 1, \ldots, N$ at time index k and $\mathbf{s}(k)$ the $(1 \times p)$ external quaternion-valued input vector. The overall input to the network $\mathbf{z}(k)$ represents the concatenation of vectors $\mathbf{y}(k)$, $\mathbf{s}(k)$ and the bias input $(1 + \imath + \jmath + \kappa)$, and is given by

$$\begin{aligned}
\mathbf{z}(k) &= [s(k\text{–}1), \ldots, s(k\text{–}p), 1\!+\!\imath\!+\!\jmath\!+\!\kappa, y_1(k\text{–}1), \ldots, y_N(k\text{–}1)]^T \\
&= z_n^a(k) + z_n^b(k)\imath + z_n^c(k)\jmath + z_n^d(k)\kappa
\end{aligned} \tag{12}$$

where z_n^a, z_n^b, z_n^c and z_n^d are the real-valued input components corresponding to the n^{th} element from the input vector $\mathbf{z}(k)$.

A quaternion-valued weight matrix of the network is denoted by $\mathbf{W}$, where for l^{th} neuron, we have $\mathbf{w}_l = [w_{l,1}, \ldots, w_{l,p+F+1}]^T$. In the following subsections, we shall only consider the output from the first neuron $y_1(k)$ (recurrent perceptron) which will result in the cost function of

$$E(k) = e_1(k)e_1^*(k) = (e_1^a(k))^2 + (e_1^b(k))^2 + (e_1^c(k))^2 + (e_1^d(k))^2 \tag{13}$$

where the error $e_1(k) = d(k) - y_1(k)$ with $d(k)$ being the desired signal. The terms e_1^a, e_1^b, e_1^c and e_1^d denote the error component in the real part, $\imath$ part, $\jmath$ part and κ part.

5.1. Derivation of the Split Quaternion-valued RTRL

The split Quaternion-valued Real-Time Recurrent Learning (QRTRL) algorithm for FCRNN utilizes the split-quaternion function, whose output at the l^{th} neuron $y_l(k)$ is given by

$$\begin{aligned}
y_l(k) &= \Phi_s\big(\mathbf{w}_l^T(k)\mathbf{z}(k)\big) = \Phi_s\big(\mathrm{net}_l(k)\big) \\
&= \Phi_a\big(\mathrm{net}_l^a(k)\big)\!+\!\Phi_b\big(\mathrm{net}_l^b(k)\big)\imath\!+\!\Phi_c\big(\mathrm{net}_l^c(k)\big)\jmath\!+\!\Phi_d\big(\mathrm{net}_l^d(k)\big)\kappa
\end{aligned} \tag{14}$$

where $\Phi_s(\cdot)$ denotes the "split" quaternion nonlinearity, Φ_a is a real-valued nonlinear activation function applied to the real part of net_l, Φ_b to the $\imath$ part, Φ_c to the $\jmath$ part and Φ_d to the κ part. The terms net_l^a, net_l^b, net_l^c and net_l^d are real-valued, and are given by [6]

$$\begin{bmatrix} \mathrm{net}_l^a \\ \mathrm{net}_l^b \\ \mathrm{net}_l^c \\ \mathrm{net}_l^d \end{bmatrix} = \begin{bmatrix} (\mathbf{w}_l^a)^T\mathbf{z}^a - (\mathbf{w}_l^b)^T\mathbf{z}^b - (\mathbf{w}_l^c)^T\mathbf{z}^c - (\mathbf{w}_l^d)^T\mathbf{z}^d \\ (\mathbf{w}_l^a)^T\mathbf{z}^b + (\mathbf{w}_l^b)^T\mathbf{z}^a + (\mathbf{w}_l^c)^T\mathbf{z}^d - (\mathbf{w}_l^d)^T\mathbf{z}^c \\ (\mathbf{w}_l^a)^T\mathbf{z}^c + (\mathbf{w}_l^c)^T\mathbf{z}^a + (\mathbf{w}_l^d)^T\mathbf{z}^b - (\mathbf{w}_l^b)^T\mathbf{z}^d \\ (\mathbf{w}_l^a)^T\mathbf{z}^d + (\mathbf{w}_l^d)^T\mathbf{z}^a + (\mathbf{w}_l^b)^T\mathbf{z}^c - (\mathbf{w}_l^c)^T\mathbf{z}^b \end{bmatrix} \tag{15}$$

The split QRTRL then minimizes the cost function (13) through a gradient descent weight update specified by $w_{s,t}(k+1) = w_{s,t}(k) - \mu\nabla_{w_{s,t}}E(k)$ where μ is the real-valued learning rate and the gradient $\nabla_{w_{s,t}}E(k)$ is given by

$$\nabla_{w_{s,t}}E(k) = \frac{\partial E(k)}{\partial w_{s,t}^*(k)} = \frac{\partial E(k)}{\partial w_{s,t}^a(k)} + \frac{\partial E(k)}{\partial w_{s,t}^b(k)}\imath + \frac{\partial E(k)}{\partial w_{s,t}^c(k)}\jmath + \frac{\partial E(k)}{\partial w_{s,t}^d(k)}\kappa \tag{16}$$

We first expand the term $\frac{\partial E}{\partial w_{s,t}^a}$ in (16) based on the cost function in (13) to give

$$\frac{\partial E(k)}{\partial w_{s,t}^a(k)} = -e_1^a(k)\frac{\partial y_l^a(k)}{\partial w_{s,t}^a(k)} - e_1^b(k)\frac{\partial y_l^b(k)}{\partial w_{s,t}^a(k)} - e_1^c(k)\frac{\partial y_l^c(k)}{\partial w_{s,t}^a(k)} - e_1^d(k)\frac{\partial y_l^d(k)}{\partial w_{s,t}^a(k)} \tag{17}$$

where the terms $\frac{\partial y_l^a}{\partial w_{s,t}^a}$, $\frac{\partial y_l^b}{\partial w_{s,t}^a}$, $\frac{\partial y_l^c}{\partial w_{s,t}^a}$ and $\frac{\partial y_l^d}{\partial w_{s,t}^a}$ represents the real-valued sensitivity of the network. For convenience, we denote the sensitivity terms in (17) with $\Psi_{s,t}^{l,(\eta a)}=\frac{\partial y_l^\eta}{\partial w_{s,t}^a}$ where $\eta \in \{a,b,c,d\}$, resulting in

$$\frac{\partial E(k)}{\partial w_{s,t}^a(k)}=-e_1^a(k)\Psi_{s,t}^{l,(aa)}(k)-e_1^b(k)\Psi_{s,t}^{l,(ba)}(k)-e_1^c(k)\Psi_{s,t}^{l,(ca)}(k) -e_1^d(k)\Psi_{s,t}^{l,(da)}(k) \quad (18)$$

In order to make further calculations feasible, we assume a small stepsize so that [5, 10]

$$\begin{aligned}\mathbf{w}(k)\approx\mathbf{w}(k-1)\approx\cdots\approx\mathbf{w}(k-M)\\ \frac{\partial y(k)}{\partial \mathbf{w}(k)}\approx\frac{\partial y(k)}{\partial \mathbf{w}(k-1)}\approx\cdots\approx\frac{\partial y(k)}{\partial \mathbf{w}(k-M)}\end{aligned} \quad (19)$$

We begin calculating the sensitivity $\Psi_{s,t}^{l,(aa)}(k)$ by differentiating $y_l^a(k)$ w.r.t. $w_{s,t}^a$ and applying the assumptions in (19) to yield

$$\begin{aligned}\Psi_{s,t}^{l,aa}(k)&=\frac{\partial y_l^a(k)}{\partial \mathrm{net}_l^a(k)}\frac{\partial \mathrm{net}_l^a(k)}{\partial w_{s,t}^a(k)}\\ &=\Phi_s^{'}\big(\mathrm{net}_l^a(k)\big)\bigg(\delta_{sl}z_l^a(k)+\sum_{q=1}^{N}\frac{\partial y_l(k-1)}{\partial w_{s,t}^a(k)}\bigg)\\ &=\Phi_s^{'}\big(\mathrm{net}_l^a(k)\big)\bigg(\delta_{sl}z_l^a(k)+\sum_{q=1}^{N}w_{l,p+1+q}^a(k)\Psi_{s,t}^{q,(aa)}(k-1)\\ &-w_{l,p+1+q}^b(k)\Psi_{s,t}^{q,(ba)}(k-1)-w_{l,p+1+q}^c(k)\Psi_{s,t}^{q,(ca)}(k-1)\\ &-w_{l,p+1+q}^d(k)\Psi_{s,t}^{q,(da)}(k-1)\bigg)\end{aligned} \quad (20)$$

Following a similar approach, we may derive recursive equations for the other 15 sensitivity terms. We can then group these sensitivity terms together to obtain the compact solution of

$$\mathbf{\Psi}_{s,t}^l(k)=\mathbf{\Phi_s}^{'}(k)\bigg(\sum_{q=1}^{N}\mathbf{W}(k)\mathbf{\Psi}_{s,t}^q(k-1)+\delta_{sl}\mathbf{z}_{\mathrm{split}}(k)\bigg) \quad (21)$$

where δ_{sl} is the dirac-delta function. Each of the real-valued matrices are given as

$$\mathbf{\Psi}_{s,t}^l=\begin{bmatrix}\Psi_{s,t}^{l,(aa)} & \Psi_{s,t}^{l,(ab)} & \Psi_{s,t}^{l,(ac)} & \Psi_{s,t}^{l,(ad)}\\ \Psi_{s,t}^{l,(ba)} & \Psi_{s,t}^{l,(bb)} & \Psi_{s,t}^{l,(bc)} & \Psi_{s,t}^{l,(bd)}\\ \Psi_{s,t}^{l,(ca)} & \Psi_{s,t}^{l,(cb)} & \Psi_{s,t}^{l,(cc)} & \Psi_{s,t}^{l,(cd)}\\ \Psi_{s,t}^{l,(da)} & \Psi_{s,t}^{l,(db)} & \Psi_{s,t}^{l,(dc)} & \Psi_{s,t}^{l,(dd)}\end{bmatrix}$$

$$\mathbf{\Phi_s}^{'}=\begin{bmatrix}\Phi_a^{'}(\mathrm{net_l^a}) & 0 & 0 & 0\\ 0 & \Phi_b^{'}(\mathrm{net_l^b}) & 0 & 0\\ 0 & 0 & \Phi_c^{'}(\mathrm{net_l^c}) & 0\\ 0 & 0 & 0 & \Phi_d^{'}(\mathrm{net_l^d})\end{bmatrix}$$

$$\mathbf{W}=\begin{bmatrix}w_{l,p+1+q}^a & -w_{l,p+1+q}^b & -w_{l,p+1+q}^c & -w_{l,p+1+q}^d\\ w_{l,p+1+q}^a & w_{l,p+1+q}^b & w_{l,p+1+q}^c & -w_{l,p+1+q}^d\\ w_{l,p+1+q}^a & -w_{l,p+1+q}^b & w_{l,p+1+q}^c & w_{l,p+1+q}^d\\ w_{l,p+1+q}^a & w_{l,p+1+q}^b & -w_{l,p+1+q}^c & w_{l,p+1+q}^d\end{bmatrix}$$

$$\mathbf{z}_{\mathrm{split}}=\begin{bmatrix}z_l^a & z_l^b & z_l^c & z_l^d\\ -z_l^b & z_l^a & -z_l^d & z_l^c\\ -z_l^c & z_l^d & z_l^a & -z_l^b\\ -z_l^d & -z_l^c & z_l^b & z_l^a\end{bmatrix} \quad (22)$$

5.2. Derivation of the Quaternion-valued RTRL

For fully Quaternion-valued RTRL, the ouptut $y_l(k)$ is given by

$$y_l(k)=\Phi\big(\mathbf{w}_l^T(k)\mathbf{z}(k)\big)=\Phi\big(\mathrm{net}_l(k)\big) \quad (23)$$

We shall first express the gradient $\nabla_{\mathbf{w}}E(k)$ as

$$\nabla_{\mathbf{w}}E(k)=e_1(k)\frac{\partial e_1^*(k)}{\partial \mathbf{w}^*}+\frac{\partial e_1(k)}{\partial \mathbf{w}^*}e_1^*(k)=-e_1(k)\mathbf{\Upsilon}(k)-\mathbf{\Psi}(k)e_1^*(k) \quad (24)$$

where $\mathbf{\Upsilon}(k)$ and $\mathbf{\Psi}(k)$ are the conjugate sensitivities and sensitivities respectively, defined by

$$\begin{aligned}\mathbf{\Psi}(k) &= \bigg[\frac{\partial y_1(k)}{\partial w_{1,1}(k)},\ldots,\frac{\partial y_l(k)}{\partial w_{N,N+p+1}(k)}\bigg]\\ \mathbf{\Upsilon}(k) &= \bigg[\frac{\partial (y_1)^*(k)}{\partial w_{1,1}(k)},\ldots,\frac{\partial (y^l)^*(k)}{\partial w_{N,N+p+1}(k)}\bigg]\end{aligned} \quad (25)$$

Calculate the sensitivity $\Psi_{s,t}^l(k)$ by differentiating $y_l(k)$ in (23) w.r.t. $w_{s,t}$, resulting in

$$\Psi_{s,t}^l(k)=\frac{\partial y_l(k)}{\partial w_{s,t}^a(k)}+\frac{\partial y_l(k)}{\partial w_{s,t}^b(k)}\imath+\frac{\partial y_l(k)}{\partial w_{s,t}^c(k)}\jmath+\frac{\partial y_l(k)}{\partial w_{s,t}^d(k)}\kappa \quad (26)$$

To find the term $\frac{\partial y^l}{\partial w_{s,t}^a}$, differentiate y^l w.r.t. $w_{s,t}^a$ to yield

$$\begin{aligned}\frac{\partial y_l(k)}{\partial w_{s,t}^a(k)}&=\frac{\partial y_l(k)}{\partial \mathrm{net}_l(k)}\frac{\partial \mathrm{net}_l(k)}{\partial w_{s,t}^a(k)}\\ &=\Phi^{'}(\mathrm{net}_l(k))\bigg(\delta_{sl}\big(z_l^a(k)+z_l^b(k)\imath+z_l^c(k)\jmath+z_l^d(k)\kappa\big)\\ &+\sum_{q=1}^{N}w_{s,t}(k)\frac{\partial y_q(k-1)}{\partial w_{s,t}^a(k)}\bigg)\end{aligned} \quad (27)$$

Similar to the derivation of $\frac{\partial y_l}{\partial w_{s,t}^a}$, the terms $\frac{\partial y_l}{\partial w_{s,t}^b}$, $\frac{\partial y_l}{\partial w_{s,t}^c}$ and $\frac{\partial y_l}{\partial w_{s,t}^d}$ can also be found. Upon summing up these terms and applying the assumption of small weight change, this yields

$$\Psi_{s,t}^l(k)=\Phi^{'}(\mathrm{net}_l(k))\bigg(-2\delta_{sl}z_l^*(k)+\sum_{q=1}^{N}w_{s,t}(k)\Psi_{s,t}^q(k-1)\bigg) \quad (28)$$

Proceeding in a similar manner, the expression for $\Upsilon_{s,t}^l$ becomes

$$\Upsilon_{s,t}^l(k)=\Phi^{'*}(\mathrm{net}_l(k))\bigg(4\delta_{sl}z_l^*(k)+\sum_{q=1}^{N}\Upsilon_{s,t}^q(k-1)w_{s,t}^*(k)\bigg) \quad (29)$$

It is clear that we only need two quaternion-valued sensitivities $\mathbf{\Upsilon}$ and $\mathbf{\Psi}$ in (28) - (29) to govern the system in contrast with the 16 real-valued sensitivities in the split-quaternion case shown in (22), resulting in a reduced computational complexity.

6. SIMULATIONS

The $\tanh(q)$ was chosen as the nonlinear activation function and initial values for $\mathbf{\Psi}(k)$ and $\mathbf{\Upsilon}(k)$ were set to zero for both algorithms. The algorithms had the input tap length of $p=3$ and output neurons of $N=2$; the performance was assessed in a predictive setting. For simulation purposes, the three-dimensional Lorenz chaotic signal [10] and three-dimensional real-world Tai Chi motion recorded from 3D inertial motion sensors were considered.

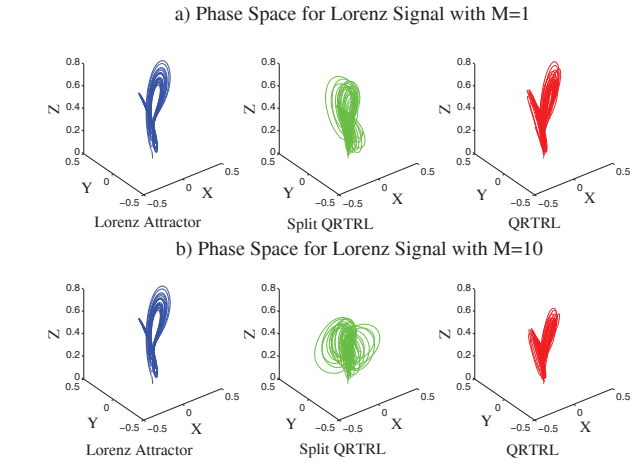

Fig. 1. Phase space of Lorenz signal

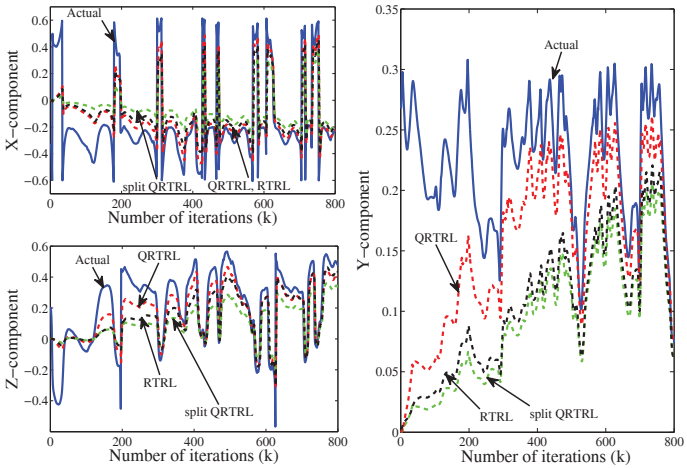

Fig. 2. The performance of QRTRL, split QRTRL and RTRL on the prediction of motion data

6.1. Three-dimensional Lorenz Chaotic Signal

The Lorenz attractor is a three-dimensional chaotic system used to model atmospheric turbulence with the governing equations given in [10]. For this experiment, the learning rate for both algorithms were set to $\mu = 5 \times 10^{-4}$. Figure 1a shows the original Lorenz attractor and the reconstruction of the attractor in the phase space for both algorithms for one step ahead ($M = 1$) prediction. Although both algorithms were able to reconstruct the attractor, the QRTRL estimated a more accurate replica of the attractor than the split QRTRL.

Figure 1b depicts the Lorenz attractor and the reconstructed attractor for both algorithms for ten steps ahead prediction ($M = 10$). It is apparent that the output of the QRTRL still resembles the original Lorenz attractor, therefore outperforming the split QRTRL.

6.2. Motion Estimation

Five 3D gyroscopes were placed on the left arm, left hand, right arm, right hand and the waist of an athlete performing Tai Chi movements and 3D motion data were recorded using the XSense MTx 3DOF Orientation Tracker. The movement of the left arm was used as a pure quaternion input for this simulation. For a fair comparison, we also considered the performance of three parallel real-valued RNNs trained with RTRL [10]. The learning rate was set to $\mu = 1 \times 10^{-3}$ for the quaternion-valued algorithms and $\mu = 1 \times 10^{-2}$ for the RTRL since it performed poorly at smaller learning rate.

Figure 2 shows the componentwise performance of the one step ahead prediction $M = 1$ of the Tai Chi motion using the split QRTRL, QRTRL and RTRL algorithms. It can be seen that the algorithms performed similarly in the X-component. However, the QRTRL outperformed the split QRTRL and RTRL in Y- and Z-components. The performance for both the split QRTRL and RTRL were similar for all three-dimensions. The performances of the QMLP for both experiments were identical to the split QRTRL and were omitted due to space limitations.

7. CONCLUSION

A class of quaternion-valued learning algorithms has been introduced. This has been achieved by using a local gradient in [8] for nonlinear adaptive filters with feedback. The superior performances of the fully-quaternion algorithm (QRTRL) compared to the split-quaternion algorithm (split QRTRL) stems from the fact that QRTRL accounts for the interchannel couplings in contrast with the split QRTRL. The componentwise channel processing which operates on both split QRTRL and RTRL explains their similar performances. Simulations over the chaotic Lorenz attractor and real world three-dimensional motion data illustrate the advantages of the proposed approach. The same framework can be used to introduce any other nonlinear gradient based learning algorithm, and by removing the nonlinearity, learning algorithms for IIR filters are obtained.

8. REFERENCES

[1] S. Said, N. L. Bihan, and S. J. Sangwine, "Fast complexified quaternion Fourier transform," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1522–1531, 2008.

[2] N. L. Bihan and J. Mars, "Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing," *Signal Processing*, vol. 8, pp. 1177–1199, 2004.

[3] J. Via, D. Ramirez, and I. Santamaria, "Properness and widely linear processing of quaternion random vectors," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3502–3515, 2010.

[4] C. Cheong Took and D. P. Mandic, "The quaternion LMS algorithm for adaptive filtering of hypercomplex processes," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1316–1327, 2009.

[5] C. Cheong Took and D. P. Mandic, "Quaternion-valued stochastic gradient-based adaptive IIR filtering," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3895–3901, 2010.

[6] B. Che Ujang, C. Cheong Took, and D. P. Mandic, "Quaternion valued nonlinear adaptive filtering," *IEEE Transactions on Neural Networks*, vol. 22, no. 8, pp. 1193–1206, 2011.

[7] T. Kim and T. Adali, "Approximation by fully complex multilayer perceptrons," *Neural Computation*, vol. 15, no. 7, pp. 1641–1666, 2003.

[8] S. De Leo and P. Rotelli, "Quaternion analyticity," *Applied Mathematics Letters*, vol. 16, no. 7, pp. 1077–1081, 2003.

[9] D. P. Mandic and V. S. L. Goh, *Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models*, Wiley, 2009.

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