The Widely Linear Quaternion Recursive Least Squares Filter

Cyrus Jahanchahi, Clive Cheong Took and Danilo P. Mandic
Communications and Signal Processing Research Group
Imperial College London, London
Email: {cyrus.jahanchahi, c.cheong-took, d.mandic}@imperial.ac.uk

***Abstract*—A quaternion valued recursive least squares algorithm for the processing of the generality of quaternion valued random processes (both circular and noncircular) is introduced. This is achieved by extending the widely linear model from the complex domain, and accounting for the specific properties of quaternion algebra. Firstly, the widely linear quaternionic Wiener solution is introduced which uses the 'augmented' input and weight vectors and thus makes full use of the available second order information. Next, the widely linear quaternion recursive least squares (WL-QRLS) algorithm is derived and is shown to exhibit enhanced transient and steady state properties as compared to the standard widely linear quaternion least mean square (WL-QLMS). Simulations on real world 3D wind signal support the approach.**

I. Introduction

The recursive least squares (RLS) algorithm is a standard in many real world applications requiring accurate adaptive filters. Compared to the least mean square (LMS) algorithm, it generally exhibits improved convergence and better steady state properties, at the cost of greater computational requirement. It also has a desirable property that the rate of convergence is invariant to the condition number of the correlation matrix R of the input signal [1].

The recent advances in sensing technology [2], robotics [3] and human centered computing have brought to light problems involving vector sensors, which are typically three- and four-dimensional. This has resulted in the development of the corresponding signal processing algorithms, making them suitable for the operation directly in the domain where the observed processes reside. One such example is the recent development of statistical signal processing for quaternion valued signals, where the power of quaternion algebra has been shown to be advantageous in the processing of three- and four-dimensional signals, such as color images [4], bodysensor measurements [5], communications and renewable energy [6].

We have also witnessed the development of adaptive signal processing algorithms suitable for the quaternion domain, thus allowing a unified filtering of three- and four-dimensional signals [7]. In analogy to the complex domain, where in order to account for second order complex noncircularity (improperness), a whole class of adaptive filtering algorithms suitable for the generality of complex signals (both proper and improper) has been developed [8], the first step in this direction in the quaternion domain is the widely-linear QLMS (WL-QLMS) algorithm [9]. It is important to notice that whereas the so called widely linear modeling in $\mathbb{C}$ is based on combining the complex vector and its complex conjugate to produce a new, 'augmented' input, the situation in the quaternion domain is radically different, not only due to the higher dimensionality, but also owing to the special properties of quaternion algebra.

To illustrate the need for widely linear quaternion modeling, consider the mean square error estimator (MSE), which estimates a variable y in terms of an observation x. The estimate $\hat{y}$ that minimizes the MSE error is the conditional expectation

$$\hat{y} = E[y|x] \tag{1}$$

which for the zero mean, jointly normal y and x, becomes a linear model

$$\hat{y} = \boldsymbol{\alpha}^T \mathbf{x} \tag{2}$$

where $\boldsymbol{\alpha}$ is a vector of coefficients, and $\mathbf{x} = [x_1, ..., x_N]^T$. In the complex domain, the estimator in (2) is only valid for circular signals, and an enhanced model, called the widely linear model must be used [8]. To show this, denote the real and imaginary parts of complex quantities by $(\cdot)_r$ and $(\cdot)_i$. Then,

$$\hat{y} = E[y_r|x_r, x_i] + jE[y_i|x_r, x_i]$$

Given $x_r = \frac{x+x^*}{2}$ and $x_i = \frac{x-x^*}{2j}$, we have

$$\hat{y} = E[y|\mathbf{x}, \mathbf{x}^*]$$

that is, for an optimal linear estimator, the 'augmented' input $[\mathbf{x}^T, \mathbf{x}^H]^T$ must be used, leading to the widely linear model [8]

$$\hat{y} = \boldsymbol{\alpha}^T \mathbf{x} + \boldsymbol{\beta}^T \mathbf{x}^* \tag{3}$$

Analogously, for a quaternion variable $q = q_a + iq_b + jq_c + kq_d$, the linear MSE estimator has the components

$$\hat{y}_m = E[y_m|\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d], \quad m \in \{a, b, c, d\}$$

and the corresponding quaternion widely model becomes

$$\hat{y} = \mathbf{u}^T \mathbf{x} + \mathbf{v}^T \mathbf{x}^i + \mathbf{g}^T \mathbf{x}^j + \mathbf{h}^T \mathbf{x}^k \tag{4}$$

as explained in Section IV.

In this work, we first develop the widely linear quaternion model and proceed with the derivation of the widely linear quaternion RLS (WL-QRLS), showing that it offers enhanced performance for the general class of noncircular signals. The enhanced performance of the WL-QRLS algorithm is illustrated by simulations which compare the WL-QLMS and WL-QRLS on the prediction of four dimensional signals, such as real-world wind measurements.

II. Quaternion Algebra

Quaternions are an extension of complex numbers (forming an ordered pair) comprising of a real part (denoted by a subscript a) and three imaginary parts (denoted by subscripts b, c and d). A quaternion variable $q \in \mathbb{H}$ can be described as:

$$q = q_a + iq_b + jq_c + kq_d \tag{5}$$

The unit axis vectors i, j and k in the quaternion domain $\mathbb{H}$ are also imaginary units, and obey the following rules

$$ij = k \quad jk = i \quad ki = j$$

$$i^2 = j^2 = k^2 = ijk = -1$$

Note that quaternion multiplication is not commutative, that is $ij = k \neq ji = -k$.

A quaternion variable q can be conveniently written as [10]

$$q = Sq + Vq$$

978-1-4244-6459-3/10/$26.00 ©2010 IEEE

where, $Sq = q_a$ (denotes the scalar part of q) and $Vq = iq_b + jq_c + kq_d$ (denotes the vector part of q). Then, the quaternion product can be expressed as:

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

where symbol '$\bullet$' denotes the dot-product and '$\times$' denotes the cross-product in vector analysis. The quaternion conjugate, denoted by q^* is given by

$$q^* = Sq - Vq$$

The norm $\|q\|$ of a quaternion variable q, is defined as

$$\|q\| = \sqrt{qq^*} = \sqrt{q_a^2 + q_b^2 + q_c^2 + q_d^2}$$

The three-dimensional vector part Vq is also called a pure quaternion, whereas the inclusion of the real part Sq gives a full quaternion. The unique algebraic structure of quaternions enables unified processing of three- and four-dimensional multivariate processes.

A. Quaternion Involution

Involutions [11] are self-inverse mappings and are defined as[1]

$$q^i = -iqi = q_a + iq_b - jq_c - kq_d \tag{6}$$

$$q^j = -jqj = q_a - iq_b + jq_c - kq_d \tag{7}$$

$$q^k = -kqk = q_a - iq_b - jq_c + kq_d \tag{8}$$

To verify that involutions represent self-inverse mappings, consider for instance $(q^i)^i = q$. The involution of a product is also a product of the individual involutions (i.e. $(q_1 q_2)^i = q_1^i q_2^i$). It is important to realize that involutions can be seen as a quaternionic counterpart of the complex conjugate, as they allow the components of a quaternion variable to be expressed in terms of the actual variable and its involutions, that is

$$q_a = \frac{1}{4}[q + q^i + q^j + q^k] \tag{9}$$

$$q_b = \frac{1}{4i}[q + q^i - q^j - q^k] \tag{10}$$

$$q_c = \frac{1}{4j}[q - q^i + q^j - q^k] \tag{11}$$

$$q_d = \frac{1}{4k}[q - q^i - q^j + q^k] \tag{12}$$

The above representation is instrumental in the derivation of quaternion valued widely-linear adaptive filtering models.

III. Quaternion Gradient

The cost function within the Mean Square Error (MSE) optimization in the quaternion domain is the standard error power, given by

$$J = ee^* = e_a^2 + e_b^2 + e_c^2 + e_d^2 \tag{13}$$

To obtain the Minimum MSE (MMSE) in linear estimation problems, the gradient of the cost function J is usually calculated using so called pseudogradients, or by optimizing the error powers channel-wise [12]. However, both these approaches lead to suboptimal solutions, as the error gradient should be calculated rigorously, by taking into account the non-commutativity of the quaternion product [7]. In the complex domain, it has been established that the gradient of real functions with respect to complex filter coefficients should be calculated with respect to the conjugate of the complex weights [13] [8]. In order to provide a corresponding formalism in the quaternion domain, we shall make use of $\mathbb{HR}$-calculus [14].

The $\mathbb{HR}$ derivatives with respect to the quaternion variable q and its conjugate q^* are given in (14) and (15). In Section VI we illustrate a solution to a minimization problem, based on the conjugate gradient in (15).

IV. Widely Linear Quaternion Modeling

In the complex domain, noncircular[2] signals have non-vanishing pseudocovariance $E[\mathbf{x}\mathbf{x}^T]$ [15], and for complete second order modeling of general, noncircular signals, both the pseudocovariance $E[\mathbf{x}\mathbf{x}^T]$ as well as the covariance matrix $E[\mathbf{x}\mathbf{x}^H]$ must be used. In adaptive filtering problems this translates to deriving algorithms based on the widely linear model (3).

Recently, widely linear modeling has been extended to the quaternion domain [16], where it is shown that to entirely describe the second-order statistics of quaternion noncircular random variables, the additional complementary covariance matrices $E[\mathbf{q}\mathbf{q}^{iH}]$, $E[\mathbf{q}\mathbf{q}^{jH}]$ and $E[\mathbf{q}\mathbf{q}^{kH}]$ must be employed. Analogously to the complex domain, for a noncircular process, the model should comprise the terms q, q^i, q^j and q^k, to fully capture the so called augmented statistics.

In adaptive signal processing algorithms, we obtain an estimate $y(n)$ of a teaching signal $d(n)$ in terms of the tap input $\mathbf{x}(n)$. That is,

$$y(n) = E[d(n)|\mathbf{x}(n)]$$

Based on (1)-(4), for a quaternion valued adaptive filter, we can split $y(n)$ and $\mathbf{x}(n)$ into their components to give

$$\begin{aligned} y_a(n) &= E[d_a(n)|\mathbf{x}_a(n), \mathbf{x}_b(n), \mathbf{x}_c(n), \mathbf{x}_d(n)] \\ y_b(n) &= E[d_b(n)|\mathbf{x}_a(n), \mathbf{x}_b(n), \mathbf{x}_c(n), \mathbf{x}_d(n)] \\ y_c(n) &= E[d_c(n)|\mathbf{x}_a(n), \mathbf{x}_b(n), \mathbf{x}_c(n), \mathbf{x}_d(n)] \\ y_d(n) &= E[d_d(n)|\mathbf{x}_a(n), \mathbf{x}_b(n), \mathbf{x}_c(n), \mathbf{x}_d(n)] \end{aligned} \tag{16}$$

From equations (9) to (12), the terms $\mathbf{x}_a(n)$, $\mathbf{x}_b(n)$, $\mathbf{x}_c(n)$, $\mathbf{x}_d(n)$ can be written in terms of $\mathbf{x}(n)$, $\mathbf{x}^i(n)$, $\mathbf{x}^j(n)$ and $\mathbf{x}^k(n)$, and the four expressions in (16) can be rewritten as:

$$\begin{aligned} y_a(n) &= E[d_a(n)|\mathbf{x}(n), \mathbf{x}^i(n), \mathbf{x}^j(n), \mathbf{x}^k(n)] \\ y_b(n) &= E[d_b(n)|\mathbf{x}(n), \mathbf{x}^i(n), \mathbf{x}^j(n), \mathbf{x}^k(n)] \\ y_c(n) &= E[d_c(n)|\mathbf{x}(n), \mathbf{x}^i(n), \mathbf{x}^j(n), \mathbf{x}^k(n)] \\ y_d(n) &= E[d_d(n)|\mathbf{x}(n), \mathbf{x}^i(n), \mathbf{x}^j(n), \mathbf{x}^k(n)] \end{aligned}$$

and the output $y(n)$ of a quaternion valued FIR filter can be expressed as

$$y(n) = E[d(n)|\mathbf{x}(n), \mathbf{x}^i(n), \mathbf{x}^j(n), \mathbf{x}^k(n)] \tag{17}$$

In other words, to capture full second order information available we should use the adaptive version of the widely linear quaternionic model (4), to give the filter output

$$y(n) = \mathbf{w}^T(n)\mathbf{q}(n) \tag{18}$$

where the augmented weight vector $\mathbf{w}(n)$ and input vector $\mathbf{q}(n)$, are defined respectively as

$$\mathbf{q}(n) = [\mathbf{x}^T(n) \;\; \mathbf{x}^{iT}(n) \;\; \mathbf{x}^{jT}(n) \;\; \mathbf{x}^{kT}(n)]^T \tag{19}$$

$$\mathbf{w}(n) = [\mathbf{u}^T(n) \;\; \mathbf{v}^T(n) \;\; \mathbf{g}^T(n) \;\; \mathbf{h}^T(n)]^T \tag{20}$$

[1]Note that the quaternion conjugate is also an involution.

[2]Complex circularity refers to rotation-invariant probability distributions. A second order circular signal is termed 'proper', whereas a second order noncircular signal is termed improper.

$$\frac{\partial f(q,q^i,q^j,q^k)}{\partial q} = \frac{1}{4}\left(\frac{\partial f(q_a,q_b,q_c,q_d)}{\partial q_a} - i\frac{\partial f(q_a,q_b,q_c,q_d)}{\partial q_b} - j\frac{\partial f(q_a,q_b,q_c,q_d)}{\partial q_c} - k\frac{\partial f(q_a,q_b,q_c,q_d)}{\partial q_d}\right) \quad (14)$$

$$\frac{\partial f(q^*,q^{i*},q^{j*},q^{k*})}{\partial q^*} = \frac{1}{4}\left(\frac{\partial f(q_a,q_b,q_c,q_d)}{\partial q_a} + i\frac{\partial f(q_a,q_b,q_c,q_d)}{\partial q_b} + j\frac{\partial f(q_a,q_b,q_c,q_d)}{\partial q_c} + k\frac{\partial f(q_a,q_b,q_c,q_d)}{\partial q_d}\right) \quad (15)$$

V. The Widely-Linear QLMS

Recently, the widely-linear quaternionic model (18) has been incorporated into the QLMS algorithm [7] to yield the widely linear QLMS (WL-QLMS), described in [9]. With the output y(n) defined as in (18), the input error

$$e(n) = d(n) - y(n)$$

and the standard stochastic gradient, the adaptive weights become

$$\begin{aligned}
\mathbf{u}(n+1) &= \mathbf{u}(n) + \mu\left(\frac{1}{2}e(n)\mathbf{x}^*(n) - \frac{1}{4}\mathbf{x}(n)e^*(n)\right)\\
\mathbf{v}(n+1) &= \mathbf{v}(n) + \mu\left(\frac{1}{2}e(n)\mathbf{x}^{i*}(n) - \frac{1}{4}\mathbf{x}^i(n)e^*(n)\right)\\
\mathbf{g}(n+1) &= \mathbf{g}(n) + \mu\left(\frac{1}{2}e(n)\mathbf{x}^{j*}(n) - \frac{1}{4}\mathbf{x}^j(n)e^*(n)\right)\\
\mathbf{h}(n+1) &= \mathbf{h}(n) + \mu\left(\frac{1}{2}e(n)\mathbf{x}^{k*}(n) - \frac{1}{4}\mathbf{x}^k(n)e^*(n)\right) \quad (21)
\end{aligned}$$

The WL-QLMS has been shown to have performance advantage for the filtering of noncircular quaternion signals [7].

VI. Derivation of the Widely Linear QRLS

Within the RLS class of algorithms, the aim is to minimize the following objective function:

$$J(n) = \sum_{k=0}^{n}\lambda^{n-k} \mid e(k) \mid^2 = \sum_{k=0}^{n}\lambda^{n-k}e(k)e^*(k) \quad (22)$$

where the forgetting factor $\lambda \in (0,1]$ and the output error

$$e(k) = d(k) - \mathbf{w}_n^T\mathbf{q}(k) = d(k) - \sum_{r=0}^{p} w_n(r)q(k-r) \quad (23)$$

where p is the filter order, and $\mathbf{q}(k)$ and $\mathbf{w}_n$ denote respectively the augmented input vector at time k and the augmented weight vector at time n defined in (19) and (20).

A. The Wiener Solution

Our aim is to find a recursive solution that minimizes the objective function in (22). To this end, following $\mathbb{HR}$-calculus (15), we set the partial derivative with respect to $w_n^*(l)$ for $l = 1$ to p, to zero, to give

$$\begin{aligned}
\frac{\partial J(n)}{\partial w_n^*(l)} &= \sum_{k=0}^{n}\lambda^{n-k}\frac{\partial e(k)e(k)^*}{\partial w_n^*(l)}\\
&= \sum_{k=0}^{n}\lambda^{n-k}\left(e(k)\frac{\partial e^*(k)}{\partial w_n^*(l)} + \frac{\partial e(k)}{\partial w_n^*(l)}e^*(k)\right)
\end{aligned}$$

which yields,

$$\frac{\partial J(n)}{\partial w_n^*(l)} = \sum_{k=0}^{n}\lambda^{n-k}\frac{1}{2}q(k-l)e^*(k) - \sum_{k=0}^{n}\lambda^{n-k}e(k)q^*(k-l)$$

Substitute for $e(k)$ in (23) and set the cost function to zero to obtain[3]:

$$\begin{aligned}
\frac{\partial J(n)}{\partial w_n^*(l)} = \frac{1}{2}\sum_{k=0}^{n}\lambda^{n-k}q(k-l)\left(d^*(k) - \sum_{r=0}^{p}q^*(k-r)w_n^*(r)\right)\\
- \sum_{k=0}^{n}\lambda^{n-k}\left(d(k) - \sum_{r=0}^{p}w_n(r)q(k-r)\right)q^*(k-l) = 0
\end{aligned}$$

The terms above can be re-arranged as

$$\begin{aligned}
\frac{\partial J(n)}{\partial w_n^*(l)} &= \sum_{k=0}^{n}\sum_{r=0}^{p}\lambda^{n-k}w_n(r)q(k-r)q^*(k-l)\\
&+ \frac{1}{2}\sum_{k=0}^{n}\lambda^{n-k}q(k-l)d^*(k) - \sum_{k=0}^{n}\lambda^{n-k}d(k)q^*(k-l)\\
&- \frac{1}{2}\sum_{k=0}^{n}\sum_{r=0}^{p}\lambda^{n-k}q(k-l)q^*(k-r)w_n^*(r) = 0
\end{aligned}$$

In a more compact form, we have

$$\mathbf{w}_n^T\mathbf{R}_{qq}(n) - \left(\frac{1}{2}\mathbf{R}_{qq}\mathbf{w}_n^*\right)^T = \mathbf{r}_{dq}^T(n) - \frac{1}{2}\mathbf{r}_{qd}^T(n) \quad (24)$$

where

$$\begin{aligned}
\mathbf{R}_{qq}(n) &= \sum_{k=0}^{n}\lambda^{n-k}\mathbf{q}(k)\mathbf{q}^H(k)\\
\mathbf{r}_{qd}(n) &= \sum_{k=0}^{n}\lambda^{n-k}\mathbf{q}(k)d^*(k)\\
\mathbf{r}_{dq}(n) &= \sum_{k=0}^{n}\lambda^{n-k}d^*(k)\mathbf{q}(k) \quad (25)
\end{aligned}$$

Observe that $(\mathbf{w}_n^T\mathbf{R}_{qq}(n))^* = \left((\mathbf{w}_n^T\mathbf{R}_{qq}(n))^H\right)^T = \left(\mathbf{R}_{qq}^H(n)\mathbf{w}_n^*\right)^T = (\mathbf{R}_{qq}(n)\mathbf{w}_n^*)^T$ and $(\mathbf{r}_{dq}^T(n))^* = \mathbf{r}_{qd}^T(n)$. It therefore follows from (24) that

$$\mathbf{w}_n^T\mathbf{R}_{qq}(n) = \mathbf{r}_{dq}^T(n)$$

Post-multiplying both sides by $\mathbf{R}_{qq}^{-1}(n)$ we finally obtain

$$\mathbf{w}_n^T = \mathbf{r}_{dq}^T(n)\mathbf{R}_{qq}^{-1}(n) \quad (26)$$

The Wiener filter solution in the quaternion domain can be obtained from (26), as the recursive estimate of the correlation matrix $\mathbf{R}_{qq}(n)$ converges to a true correlation matrix $\mathbf{R}_{qq}$ for $n \to \infty$. The same applies to the estimated cross-correlation $\mathbf{r}_{dq}$, giving

$$\begin{aligned}
\mathbf{R}_{qq}^{-1}(n) &= n^{-1}\mathbf{R}_{qq}^{-1}\\
\mathbf{r}_{dq}^T(n) &= n\mathbf{r}_{dq}^T
\end{aligned}$$

and the Wiener filter solution

$$\mathbf{w}^T = \mathbf{r}_{dq}^T\mathbf{R}_{qq}^{-1} \quad (27)$$

The difference from the complex domain Wiener solution arises due to the non-commutativity of quaternion products, as the position of $\mathbf{r}_{dq}$ relative to $\mathbf{R}_{qq}^{-1}(n)$ is important and hence the term $\mathbf{w}^T$.

[3]The quaternion conjugate property $(q_1q_2)^* = q_2^*q_1^*$ was used.

We shall next derive the WL-QRLS algorithm, a recursive solution to the Wiener filtering problem in (26). Our approach follows the general idea of the derivation of the complex widely-linear RLS algorithm[4] in [17], and extends this approach by accounting for the specific properties of quaternion algebra.

B. The widely-linear QRLS algorithm

The term $\mathbf{q}(k)\mathbf{q}(k)^H$ within the covariance matrix $\mathbf{R}_{qq}$ in (25) can be expressed as

$$\begin{bmatrix} \mathbf{x}(k)\mathbf{x}^H(k) & \mathbf{x}(k)\mathbf{x}^{iH}(k) & \mathbf{x}(k)\mathbf{x}^{jH}(k) & \mathbf{x}(k)\mathbf{x}^{kH}(k) \\ \mathbf{x}^i(k)\mathbf{x}^H(k) & \mathbf{x}^i(k)\mathbf{x}^{iH}(k) & \mathbf{x}^i(k)\mathbf{x}^{jH}(k) & \mathbf{x}^i(k)\mathbf{x}^{kH}(k) \\ \mathbf{x}^j(k)\mathbf{x}^H(k) & \mathbf{x}^j(k)\mathbf{x}^{iH}(k) & \mathbf{x}^j(k)\mathbf{x}^{jH}(k) & \mathbf{x}^j(k)\mathbf{x}^{kH}(k) \\ \mathbf{x}^k(k)\mathbf{x}^H(k) & \mathbf{x}^k(k)\mathbf{x}^{iH}(k) & \mathbf{x}^k(k)\mathbf{x}^{jH}(k) & \mathbf{x}^k(k)\mathbf{x}^{kH}(k) \end{bmatrix} \tag{28}$$

yielding a simpler form

$$\mathbf{R}_{qq} = \sum_{k=1}^{n} \lambda^{n-k}\mathbf{q}(k)\mathbf{q}(k)^H = \begin{bmatrix} \mathbf{R} & \mathbf{P}^i & \mathbf{S}^j & \mathbf{T}^k \\ \mathbf{P} & \mathbf{R}^i & \mathbf{T}^j & \mathbf{S}^k \\ \mathbf{S} & \mathbf{T}^i & \mathbf{R}^j & \mathbf{P}^k \\ \mathbf{T} & \mathbf{S}^i & \mathbf{P}^j & \mathbf{R}^k \end{bmatrix} \tag{29}$$

where

$$\begin{aligned} \mathbf{R} &= \sum_{k=1}^{n} \lambda^{n-k}\mathbf{q}(k)\mathbf{q}^H(k) \\ \mathbf{P} &= \sum_{k=1}^{n} \lambda^{n-k}\mathbf{q}^i(k)\mathbf{q}^H(k) \\ \mathbf{S} &= \sum_{k=1}^{n} \lambda^{n-k}\mathbf{q}^j(k)\mathbf{q}^H(k) \\ \mathbf{T} &= \sum_{k=1}^{n} \lambda^{n-k}\mathbf{q}^k(k)\mathbf{q}^H(k) \end{aligned} \tag{30}$$

To demonstrate the matrix $\mathbf{R}_{qq}^{-1}$ has the same structure as the matrix $\mathbf{R}_{qq}$, observe that when $\mathbf{R}_{qq}^{-1}$ takes the form in (29) then $\mathbf{R}_{qq}\mathbf{R}_{qq}^{-1} = \mathbf{I}$ is a valid solution, that is

$$\begin{aligned} \mathbf{R}_{qq}\mathbf{R}_{qq}^{-1} &= \begin{bmatrix} \mathbf{R} & \mathbf{P}^i & \mathbf{S}^j & \mathbf{T}^k \\ \mathbf{P} & \mathbf{R}^i & \mathbf{T}^j & \mathbf{S}^k \\ \mathbf{S} & \mathbf{T}^i & \mathbf{R}^j & \mathbf{P}^k \\ \mathbf{T} & \mathbf{S}^i & \mathbf{P}^j & \mathbf{R}^k \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B}^i & \mathbf{C}^j & \mathbf{D}^k \\ \mathbf{B} & \mathbf{A}^i & \mathbf{D}^j & \mathbf{C}^k \\ \mathbf{C} & \mathbf{D}^i & \mathbf{A}^j & \mathbf{B}^k \\ \mathbf{D} & \mathbf{C}^i & \mathbf{B}^j & \mathbf{A}^k \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \end{aligned}$$

Out of the 16 equations (one for each entry of the identity matrix),

$$\begin{gathered} \mathbf{RA} + \mathbf{P}^i\mathbf{B} + \mathbf{S}^j\mathbf{C} + \mathbf{T}^k\mathbf{D} = \mathbf{I} \\ \mathbf{PA} + \mathbf{R}^i\mathbf{B} + \mathbf{T}^j\mathbf{C} + \mathbf{S}^k\mathbf{D} = \mathbf{0} \\ \mathbf{SA} + \mathbf{T}^i\mathbf{B} + \mathbf{R}^j\mathbf{C} + \mathbf{P}^k\mathbf{D} = \mathbf{0} \\ \mathbf{TA} + \mathbf{S}^i\mathbf{B} + \mathbf{P}^j\mathbf{C} + \mathbf{R}^k\mathbf{D} = \mathbf{0} \end{gathered}$$

only four are linearly independent, as all others are involutions of these equations. This set of linear equations has four unknowns $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$. As there is only one solution for $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$, hence the structure assumed for $\mathbf{R}_{qq}^{-1}$ must be valid. A direct consequence of this structure is that the matrix $\mathbf{R}_{qq}^{-1}$ can be fully described by only a quarter of its entries.

[4]The complex WL-RLS algorithm, proposed by C. Douglas, takes advantage of a special structure inherent to the augmented covariance matrix to simplify the calculation [17].

The first step towards the WL-QRLS algorithm is to find a recursive solution for $\mathbf{R}_{qq}^{-1}$ and $\mathbf{r}_{dq}$. Using Woodbury's identity, the matrix inverse $\mathbf{R}_{qq}^{-1}$ can be written as:

$$\begin{aligned} \mathbf{R}_{qq}^{-1}(n) &= \lambda^{-1}\mathbf{R}_{qq}^{-1}(n-1) \\ &\quad - \frac{\lambda^{-2}\mathbf{R}_{qq}^{-1}(n-1)\mathbf{q}(n)\mathbf{q}^H(n)\mathbf{R}_{qq}^{-1}(n-1)}{1+\lambda^{-1}\mathbf{q}^H(n)\mathbf{R}_{qq}^{-1}(n-1)\mathbf{q}(n)} \end{aligned} \tag{31}$$

A recursive solution for $\mathbf{A}(n)$, $\mathbf{B}(n)$, $\mathbf{C}(n)$ and $\mathbf{D}(n)$ can now be obtained by expanding the expression for $\mathbf{R}_{qq}^{-1}(n)$ into its components. Note that $\mathbf{A}(n)$, $\mathbf{B}(n)$, $\mathbf{C}(n)$ and $\mathbf{D}(n)$ can be written as

$$\begin{aligned} \mathbf{A}(n) &= \lambda^{-1}\mathbf{A}(n-1) - \lambda^{-2}\frac{\mathbf{p}\mathbf{p}^H}{c} \\ \mathbf{B}(n) &= \lambda^{-1}\mathbf{B}(n-1) - \lambda^{-2}\frac{\mathbf{p}^i\mathbf{p}^H}{c} \\ \mathbf{C}(n) &= \lambda^{-1}\mathbf{C}(n-1) - \lambda^{-2}\frac{\mathbf{p}^j\mathbf{p}^H}{c} \\ \mathbf{D}(n) &= \lambda^{-1}\mathbf{D}(n-1) - \lambda^{-2}\frac{\mathbf{p}^k\mathbf{p}^H}{c} \end{aligned} \tag{32}$$

where $\mathbf{p}$ and c are defined as

$$\begin{aligned} \mathbf{p} &= [\mathbf{A}(n-1)\mathbf{x}(n) + \mathbf{B}^i(n-1)\mathbf{x}^i(n) + \mathbf{C}^j(n-1)\mathbf{x}^j(n) \\ &\quad +\mathbf{D}^k(n-1)\mathbf{x}^k(n)] \\ c &= 1 + \lambda^{-1}4\Re\left(\mathbf{x}^H(n)\mathbf{p}\right) \end{aligned}$$

where symbol $\Re(\cdot)$ denotes the real part of a quaternion. The cross-correlation $\mathbf{r}_{dq}$ can be recursively estimated by

$$\mathbf{r}_{dq}(n) = \lambda\mathbf{r}_{dq}(n-1) + d(n)\mathbf{q}^*(n) \tag{33}$$

The augmented weight update of the WL-QRLS algorithm, $\mathbf{w}_n^T = \mathbf{r}_{dq}^T\mathbf{R}_{qq}^{-1}$ can be expanded as:

$$\begin{bmatrix} \mathbf{u}_n \\ \mathbf{v}_n \\ \mathbf{g}_n \\ \mathbf{h}_n \end{bmatrix}^T = \begin{bmatrix} \mathbf{r}_{dq_1} \\ \mathbf{r}_{dq_2} \\ \mathbf{r}_{dq_3} \\ \mathbf{r}_{dq_4} \end{bmatrix}^T \begin{bmatrix} \mathbf{A}(n) & \mathbf{B}^i(n) & \mathbf{C}^j(n) & \mathbf{D}^k(n) \\ \mathbf{B}(n) & \mathbf{A}^i(n) & \mathbf{D}^j(n) & \mathbf{C}^k(n) \\ \mathbf{C}(n) & \mathbf{D}^i(n) & \mathbf{A}^j(n) & \mathbf{B}^k(n) \\ \mathbf{D}(n) & \mathbf{C}^i(n) & \mathbf{B}^j(n) & \mathbf{A}^k(n) \end{bmatrix}$$

We can now write expressions for $\mathbf{u}_n$, $\mathbf{v}_n$, $\mathbf{g}_n$ and $\mathbf{h}_n$ in terms of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$

$$\begin{aligned} \mathbf{u}_n^T &= \mathbf{r}_{dq_1}^T\mathbf{A} + \mathbf{r}_{dq_2}^T\mathbf{B} + \mathbf{r}_{dq_3}^T\mathbf{C} + \mathbf{r}_{dq_4}^T\mathbf{D} \\ \mathbf{v}_n^T &= \mathbf{r}_{dq_1}^T\mathbf{B}^i + \mathbf{r}_{dq_2}^T\mathbf{A}^i + \mathbf{r}_{dq_3}^T\mathbf{D}^i + \mathbf{r}_{dq_4}^T\mathbf{C}^i \\ \mathbf{g}_n^T &= \mathbf{r}_{dq_1}^T\mathbf{C}^j + \mathbf{r}_{dq_2}^T\mathbf{D}^j + \mathbf{r}_{dq_3}^T\mathbf{A}^j + \mathbf{r}_{dq_4}^T\mathbf{B}^j \\ \mathbf{h}_n^T &= \mathbf{r}_{dq_1}^T\mathbf{D}^k + \mathbf{r}_{dq_2}^T\mathbf{C}^k + \mathbf{r}_{dq_3}^T\mathbf{B}^k + \mathbf{r}_{dq_4}^T\mathbf{A}^k \end{aligned} \tag{34}$$

By substituting (32) and (33) into (34), we can obtain a recursive time update for the weight vectors $\mathbf{u}_n$, $\mathbf{v}_n$, $\mathbf{g}_n$ and $\mathbf{h}_n$. For illustration, the update of the weight component $\mathbf{u}_n$ is given below. The update for the other three weight vectors are obtained analogously.

$$\begin{aligned} \mathbf{u}_n^T = &\left(\lambda\mathbf{r}_{dq_1}^T(n-1) + d(n)\mathbf{x}^H(n)\right)\left(\lambda^{-1}\mathbf{A}(n-1) - \lambda^{-2}\frac{\mathbf{p}\mathbf{p}^H}{c}\right) \\ &+ \left(\lambda\mathbf{r}_{dq_2}^T(n-1) + d(n)\mathbf{x}^{iH}(n)\right)\left(\lambda^{-1}\mathbf{B}(n-1) - \lambda^{-2}\frac{\mathbf{p}^i\mathbf{p}^H}{c}\right) \\ &+ \left(\lambda\mathbf{r}_{dq_3}^T(n-1) + d(n)\mathbf{x}^{jH}(n)\right)\left(\lambda^{-1}\mathbf{C}(n-1) - \lambda^{-2}\frac{\mathbf{p}^j\mathbf{p}^H}{c}\right) \\ &+ \left(\lambda\mathbf{r}_{dq_4}^T(n-1) + d(n)\mathbf{x}^{kH}(n)\right)\left(\lambda^{-1}\mathbf{D}(n-1) - \lambda^{-2}\frac{\mathbf{p}^k\mathbf{p}^H}{c}\right) \end{aligned}$$

Expanding $\mathbf{u}_n^T$ we can split the terms above into four groups, as shown in (35). It can be shown that the first row equals $\mathbf{u}_{n-1}^T$ and

$$
\begin{aligned}
\mathbf{u}_n^T &= \mathbf{r}_{dq_1}^T(n-1)\mathbf{A}(n-1) + \mathbf{r}_{dq_2}^T(n-1)\mathbf{B}(n-1) + \mathbf{r}_{dq_3}^T(n-1)\mathbf{C}(n-1) + \mathbf{r}_{dq_4}^T(n-1)\mathbf{A}(n-1) \\
&+ \lambda^{-1}d(n)\mathbf{x}^H(n)\mathbf{A}(n-1) + \lambda^{-1}d(n)\mathbf{x}^{iH}(n)\mathbf{B}(n-1) + \lambda^{-1}d(n)\mathbf{x}^{jH}(n)\mathbf{C}(n-1) + \lambda^{-1}d(n)\mathbf{x}^{kH}(n)\mathbf{D}(n-1) \\
&- \lambda^{-1}\mathbf{r}_{dq_1}^T(n-1)\frac{\mathbf{p}\mathbf{p}^H}{c} - \lambda^{-1}\mathbf{r}_{dq_2}^T(n-1)\frac{\mathbf{p}\mathbf{p}^{iH}}{c} - \lambda^{-1}\mathbf{r}_{dq_3}^T(n-1)\frac{\mathbf{p}\mathbf{p}^{jH}}{c} - \lambda^{-1}\mathbf{r}_{dq_4}^T(n-1)\frac{\mathbf{p}\mathbf{p}^{kH}}{c} \\
&- \lambda^{-2}d(n)\mathbf{x}^H(n)\frac{\mathbf{p}\mathbf{p}^H}{c} - \lambda^{-2}d(n)\mathbf{x}^{i^H}(n)\frac{\mathbf{p}^i\mathbf{p}^H}{c} - \lambda^{-2}d(n)\mathbf{x}^{j^H}(n)\frac{\mathbf{p}^j\mathbf{p}^H}{c} - \lambda^{-2}d(n)\mathbf{x}^{k^H}(n)\frac{\mathbf{p}^k\mathbf{p}^H}{c}
\end{aligned}
\tag{35}
$$

the second row equals $\lambda^{-1}d(n)\mathbf{p}^H(n)$. The third row can be written as

$$\lambda^{-1}\left(\mathbf{u}_{n-1}^T\mathbf{x}(n) + \mathbf{v}_{n-1}^T\mathbf{x}^i(n) + \mathbf{g}_{n-1}^T\mathbf{x}^j(n) + \mathbf{h}_{n-1}^T\mathbf{x}^k(n)\right)\frac{\mathbf{p}^H}{c}$$

whereas the last row can be written as

$$\lambda^{-2}d(n)\left(4\Re(\mathbf{x}^H\mathbf{p})\right)\frac{\mathbf{p}^H}{c}$$

Thus giving,

$$\mathbf{u}_n^T = \mathbf{u}_{n-1}^T + \lambda^{-1}d(n)\mathbf{p}^H - \lambda^{-2}d(n)(4\Re(\mathbf{x}^H(n)\mathbf{p})\frac{\mathbf{p}^H}{c} - \left(\mathbf{u}_{n-1}^T\mathbf{x}(n) + \mathbf{v}_{n-1}^T\mathbf{x}^i(n) + \mathbf{g}_{n-1}^T\mathbf{x}^j(n) + \mathbf{h}_{n-1}^T\mathbf{x}^k(n)\right)\frac{\mathbf{p}^H}{c}$$

Using $\lambda^{-1}4\Re(\mathbf{x}^H(n)\mathbf{p}) = c+1$, we finally obtain

$$
\begin{aligned}
\mathbf{u}_n^T &= \mathbf{u}_{n-1}^T + \lambda^{-1}\left(d(n) - \right. \\
&\left.\left(\mathbf{u}_{n-1}^T\mathbf{x}(n) + \mathbf{v}_{n-1}^T\mathbf{x}^i(n) + \mathbf{g}_{n-1}^T\mathbf{x}^j(n) + \mathbf{h}_{n-1}^T\mathbf{x}^k(n)\right)\right)\frac{\mathbf{p}^H}{c}
\end{aligned}
\tag{36}
$$

The update for $\mathbf{v}_n^T$, $\mathbf{g}_n^T$ and $\mathbf{h}_n^T$ follows a similar form, summarized in (43).

C. Summary of the Widely-Linear QRLS Algorithm

The WL-QRLS algorithm is summarized below

$$\mathbf{q}(n) = [\mathbf{x}^T(n)\ \mathbf{x}^{iT}(n)\ \mathbf{x}^{jT}(k)\ \mathbf{x}^{kT}(n)]^T \tag{37}$$

$$\mathbf{w}(n) = [\mathbf{u}^T(n)\ \mathbf{v}^T(n)\ \mathbf{g}^T(n)\ \mathbf{h}^T(n)]^T \tag{38}$$

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{q} \tag{39}$$

$$c = 1 + 4\Re\left(\mathbf{x}^H(n)\mathbf{p}\right) \tag{40}$$

$$
\begin{aligned}
\mathbf{p} = &[\mathbf{A}(n-1)\mathbf{x}(n) + \mathbf{B}^i(n-1)\mathbf{x}^i(n) + \mathbf{C}^j(n-1)\mathbf{x}^j(n) \\
&+\mathbf{D}^k(n-1)\mathbf{x}^k(n)]
\end{aligned}
\tag{41}
$$

$$
\begin{aligned}
\mathbf{A}(n) &= \lambda^{-1}\mathbf{A}(n-1) - \lambda^{-2}\frac{\mathbf{p}\mathbf{p}^H}{c} \\
\mathbf{B}(n) &= \lambda^{-1}\mathbf{B}(n-1) - \lambda^{-2}\frac{\mathbf{p}^i\mathbf{p}^H}{c} \\
\mathbf{C}(n) &= \lambda^{-1}\mathbf{C}(n-1) - \lambda^{-2}\frac{\mathbf{p}^j\mathbf{p}^H}{c} \\
\mathbf{D}(n) &= \lambda^{-1}\mathbf{D}(n-1) - \lambda^{-2}\frac{\mathbf{p}^k\mathbf{p}^H}{c}
\end{aligned}
\tag{42}
$$

The updates of the augmented weight vector $\mathbf{w}_n = [\mathbf{u}_n^T\ \mathbf{v}_n^T\ \mathbf{g}_n^T\ \mathbf{h}_n^T]^T$ of the WL-QRLS are given by

$$
\begin{aligned}
\mathbf{u}_n &= \mathbf{u}_{n-1} + \lambda^{-1}\frac{e(n)\mathbf{p}^H}{c} \\
\mathbf{v}_n &= \mathbf{v}_{n-1} + \lambda^{-1}\frac{e(n)\mathbf{p}^{iH}}{c} \\
\mathbf{g}_n &= \mathbf{g}_{n-1} + \lambda^{-1}\frac{e(n)\mathbf{p}^{jH}}{c} \\
\mathbf{h}_n &= \mathbf{h}_{n-1} + \lambda^{-1}\frac{e(n)\mathbf{p}^{kH}}{c}
\end{aligned}
\tag{43}
$$

VII. Simulations

The effectiveness of the WL-QRLS is demonstrated by comparing its performance to that of WL-QLMS algorithm in forecasting wind[5]. For the simulations, four signals (temperature reading and wind speed in the North, East and vertical directions[6]) were combined into one quaternion valued signal.

For comparison purposes, the step size of the widely linear QLMS algorithm [9] was chosen to give the fastest convergence for the signal at hand, whereas for the widely linear QRLS, a forgetting factor $\lambda = 0.95$ was employed. The quantitative performance measure was the prediction gain, given by

$$R_p = 10\log\frac{\sigma_x^2}{\sigma_e^2}\quad [dB] \tag{44}$$

where σ_x^2 and σ_e^2 denote respectively the input signal power and error power.

Fig. 1 compares the prediction gain of WL-QLMS and WL-QRLS for the prediction of a wind signal over a range of prediction horizons. Over all filter orders and prediction horizons, the performance of the WL-QRLS was approximately 20dB higher than that of the WL-QLMS. Fig. 2 shows the error curves for both the algorithms, illustrating the faster convergence rate of WL-QRLS. Fig. 3 compares the prediction gain when both the filters are at the steady state, illustrating the enhanced performance of the WL-QRLS. Finally, Fig. 4 visualizes the tracking performance of the WL-QRLS on the vertical wind speed component of the quaternion wind signal.

VIII. Conclusion

We have introduced the widely linear QRLS (WL-QRLS) algorithm for the processing of circular and noncircular signals in the quaternion domain. This has been achieved by extending the widely linear model to the quaternion domain, and following the RLS algorithm's steps. The superiority of the WL-QRLS has been demonstrated on the forecasting of a 4D wind field (3D wind as a pure quaternion and air temperature as a real part). The WL-QRLS algorithm has found to have an increased convergence and superior steady state performance, as compared to other available widely linear adaptive algorithms in $\mathbb{H}$, the WL-QLMS.

[5]The quaternion toolbox was used to extend the capabilities of Matlab to the quaternion domain [18].

[6]The wind data were recorded by Prof. K. Aihara and his team at the University of Tokyo, in an urban environment, sampled at 5Hz. We considered the three wind speed components as a pure quaternion, with the air temperature being the real part.

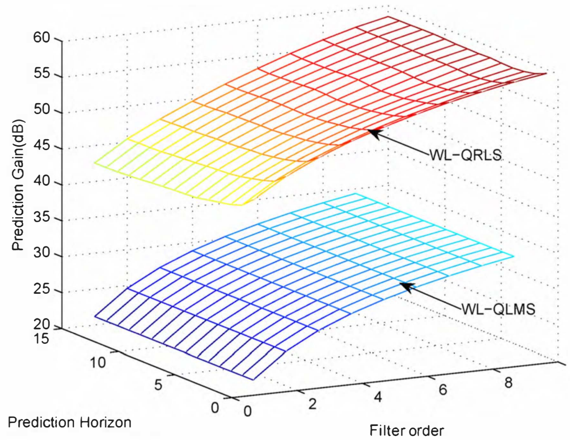

Fig. 1. Performance of WL-QLMS and WL-QRLS on the prediction of 4D wind field (3D wind speed and air temperature)

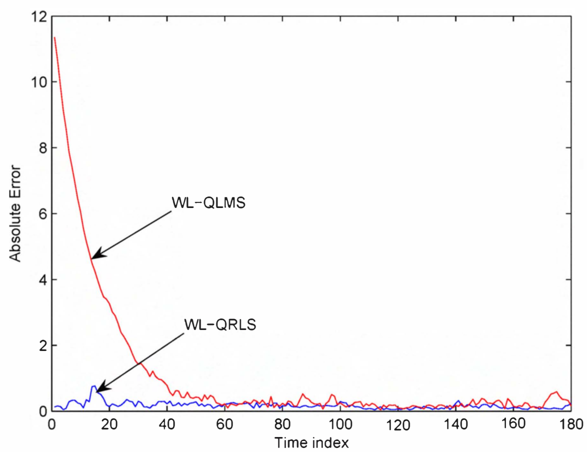

Fig. 2. Learning curves of WL-QLMS and WL-QRLS on the prediction of 4D wind field (3D wind speed and air temperature)

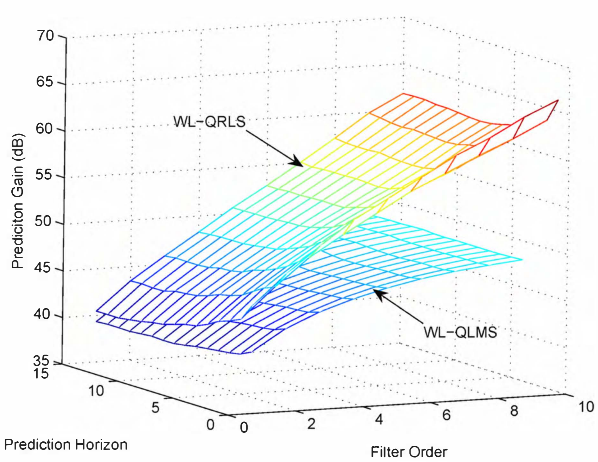

Fig. 3. Steady state performance of WL-QLMS and WL-QRLS on the prediction of 4D wind field (3D wind speed and air temperature)

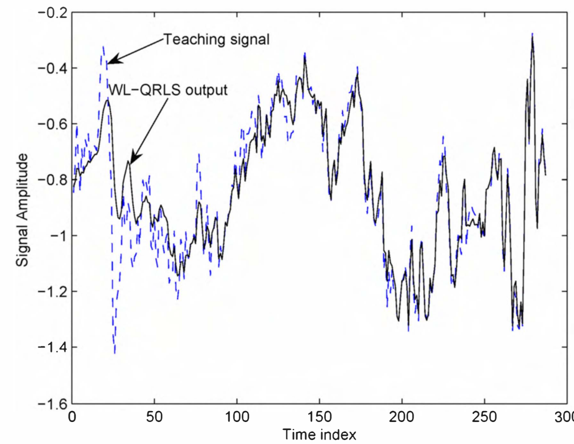

Fig. 4. Prediction performance of WL-QRLS filter for 5-step ahead prediction of the vertical wind speed component

References

[1] S. Haykin, *Adaptive Filter Theory* (fourth edition). Prentice Hall, 1996.

[2] S. Miron, N. Le Bihan, and J. Mars, "Quaternion-MUSIC for vector-sensor array processing," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1218–1229, 2006.

[3] D. Biamino, G. Cannata, M. Maggiali, and A. Piazza, "Mac-eye: A tendon driven fully embedded robot eye," in *Proc. of the 5th International Conference on Humanoid Robots*, pp. 62–67, 2005.

[4] S.-C. Pei and C.-M. Cheng, "Color image processing by using binary quaternion-moment-preserving thresholding technique," *IEEE Transactions on Image Processing*, vol. 8, no. 5, pp. 614–628, 1999.

[5] N. Rehman and D. P. Mandic, "Empirical mode decomposition for trivariate signals," *IEEE Transactions on Signal Processing*, (in print), 2010.

[6] C. Cheong-Took, D. P. Mandic, and K. Aihara, "Quaternion-valued short term forecasting of wind profile," in *Proc. of The World Congress on Computational Intelligence*, 2010.

[7] C. Cheong-Took and D. Mandic, "The quaternion LMS algorithm for adaptive filtering of hypercomplex processes," *IEEE Transactions on Signal Processing*, vol. 57, pp. 1316–1327, 2009.

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

[9] C. Cheong-Took and D. P. Mandic, "A quaternion widely linear adaptive filter," accepted in *IEEE Transactions on Signal Processing*, 2010.

[10] J. Ward, *Quaternions and Cayley numbers.* Kluwer Academic Publishers, 1997.

[11] T. A. Ella and S. J. Sangwine, "Quaternion involutions and anti-involutions," *Computers and Mathematics with Applications*, vol. 53, pp. 137–143, 2007.

[12] P. Arena, L. Fortuna, G. Muscato, and M. G. Xibilia, *Neural Networks In Multidimensional Domains: Fundamentals And New Trends In Modelling And Control.* Springer-Verlag London, 1998.

[13] K. Kreutz-Delgado, "The complex gradient operator and the $\mathbb{CR}$-calculus," Supplement to Lecture Notes, UC, San Diego, 2006.

[14] C. Jahanchahi, C. Cheong-Took, and D. P. Mandic, "$\mathbb{HR}$-calculus and quaternion analyticity," Technical Report, Imperial College London, 2010.

[15] B. Picinbono, "On circularity," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3473–3482, 1994.

[16] C. Cheong-Took and D. P. Mandic, "Augmented statistics and widely linear modelling of quaternion valued signals," Technical Report, Imperial College London, 2009.

[17] S. C. Douglas, "Widely-linear recursive least-squares algorithm for adaptive beamforming," *in Proc. of the International Conference on Acoustics, Speech and Signal Processing*, pp. 2041–2044, 2009.

[18] S. Sangwine and N. L. Bihan, Quaternion Toolbox for Matlab 2005, http://qtfm.sourceforge.net/.