This is a repository copy of *A General Quaternion-Valued Gradient Operator and Its Applications to Computational Fluid Dynamics and Adaptive Beamforming*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/94844/

Version: Accepted Version

Proceedings Paper:
Jiang, M.D., Liu, W. and Li, Y. (2014) A General Quaternion-Valued Gradient Operator and Its Applications to Computational Fluid Dynamics and Adaptive Beamforming. In: 19th International Conference on Digital Signal Processing (DSP). International Conference on Digital Signal Processing (DSP), 20-23 Aug 2014, Hong Kong. IEEE .

https://doi.org/10.1109/ICDSP.2014.6900781

Reuse
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

arXiv:1406.5719v1 [math.OC] 22 Jun 2014

A General Quaternion-valued Gradient Operator and Its Applications to Computational Fluid Dynamics and Adaptive Beamforming

Mengdi Jiang, Wei Liu
Communications Research Group
Department of Electronic and Electrical Engineering
University of Sheffield, UK
{mjiang3, w.liu}@sheffield.ac.uk

Yi Li
School of Mathematics and Statistics
University of Sheffield, S3 7RH UK
yili@sheffield.ac.uk

***Abstract*—Quaternion-valued signal processing has received increasing attention recently. One key operation involved in derivation of all kinds of adaptive algorithms is the gradient operator. Although there have been some derivations of this operator in literature with different level of details, it is still not fully clear how this operator can be derived in the most general case and how it can be applied to various signal processing problems. In this work, we will give a general derivation of the quaternion-valued gradient operator and then apply it to two different areas. One is to combine with the classic computational fluid dynamics (CFD) approach in wind profile prediction and the other one is to apply the result to the adaptive beamforming problem for vector sensor arrays.**

I. Introduction

Recently, quaternion-valued signal processing has been introduced to solve problems related to three or four-dimensional signals, such as vector-sensor array signal processing [1], [2], [3], and wind profile prediction [4], [5]. In many of the cases, the traditional complex-valued adaptive filtering operation needs to be extended to the quaternion domain to derive the corresponding adaptive algorithms. One key operation involved in derivation of quaternion-valued adaptive algorithms is the gradient operator. Although there have been some derivations of this operator in literature with different level of details, it is still not fully clear how this operator can be derived in the most general case and how it can be applied to various signal processing problems.

In this work, we will give a general derivation of the quaternion-valued gradient operator and then implement this into two different applications. One case is to combine with the classic computational fluid dynamics (CFD) approach in wind profile prediction. Wind profile prediction is a classical signal prediction problem, and we can try to solve it using traditional linear and nonlinear (neural networks) prediction techniques. On the other hand, wind/atmospheric flow analysis is also a traditional problem in CFD, which employs conservation laws, various physical models and numerical methods to predict wind signals. It might be more accurate compared to other approaches, but not without disadvantages. For example, it is time-consuming with high computational complexity, and it also contains uncertainties/errors in initial/boundary conditions as well as models. Therefore, we intend to combine the two approaches in a way that retains the efficiency of the former and the accuracy of the latter. As a preliminary study, we will apply a quaternion-valued linear predictor to the data generated by the CFD method to show the feasibility of the combined approach.

Another application of quaternion is the adaptive beamforming problem in vector sensor arrays. Adaptive beamforming has been studied extensively in the past for traditional sensor array systems [6], [7]. With the introduction of vector sensor arrays, such as those consisting of crossed-dipoles and tripoles, adaptive beamforming has been extended to this area too [8], [2]. A reference signal based adaptive beamformer will be set up employing the derived quaternion-valued least mean square (LMS) algorithm.

This paper is organized as follows. The general quaternion-valued gradient operator is derived and then applied to develop the quaternion-valued LMS (QLMS) as well as the augmented QLMS (AQLMS) algorithms in Section II. Application of the algorithms to the data generated by CFD is provided in Section III, and their application in adaptive beamforming is studied in Section IV. Simulation results are presented in Section V and conclusions are drawn in Section VI.

II. Derivation of a Quaternion-valued Gradient Operator and Adaptive Filtering

A. Differentiation with respect to a quaternion-valued vector

We first introduce the definition of differentiation with respect to a quaternion q. Assume that $f(q)$ is a function of the quaternion variable q, expressed as

$$f(q) = f_a + if_b + jf_c + kf_d \ , \tag{1}$$

where $f(q)$ is in general quaternion-valued. $f(q)$, as well as its components $f_a(q)$, $f_b(q)$, $f_c(q)$, and $f_d(q)$, can be viewed

Submitted to the 2014 International Conference on Digital Signal Processing (DSP2014, held in August 2014, Hong Kong) on 29 April 2014 and accepted on 21 May 2014.

as functions of q_a, q_b, q_c and q_d, which can be expressed in terms of q and its involutions [9]:

$$\begin{aligned} q^i &= -iqi = q_a + q_b i - q_c j - q_d k \\ q^j &= -jqj = q_a - q_b i + q_c j - q_d k \\ q^k &= -kqk = q_a - q_b i - q_c j + q_d k. \end{aligned} \tag{2}$$

As a consequence, we have

$$\begin{aligned} q_a &= \frac{1}{4}(q + q^i + q^j + q^k)\ , q_b = \frac{1}{4i}(q + q^i - q^j - q^k) \\ q_c &= \frac{1}{4j}(q - q^i + q^j - q^k)\ , q_d = \frac{1}{4k}(q - q^i - q^j + q^k) \end{aligned} \tag{3}$$

and

$$\begin{aligned} q + iqi + jqj + kqk &= -2q^* \\ q - iqi - jqj - kqk &= 4q_a. \end{aligned} \tag{4}$$

Given the above relations between the involutions and the real and imaginary parts of q, $f(q)$ can be regarded as a function of q, q^i, q^j and q^k. Therefore, in what follows, we consider generally a function of q and the involutions, i.e., $f(q, q^i, q^j, q^k)$. Using the Taylor expansion of f, the differential df is given by

$$df = \frac{\partial f}{\partial q}dq + \frac{\partial f}{\partial q^i}dq^i + \frac{\partial f}{\partial q^j}dq^j + \frac{\partial f}{\partial q^k}dq^k \tag{5}$$

Note $\partial f/\partial q$ and dq are both quaternion, therefore they do not commute. On the other hand, $df = df_a + idf_b + jdf_c + kdf_d$. Since df_a is the differential of a real function of real numbers q_a, q_b, q_c and q_d, we have

$$\begin{aligned} & df_a(q_a, q_b, q_c, q_d) \\ &= \frac{\partial f_a}{\partial q_a}dq_a + \frac{\partial f_a}{\partial q_b}dq_b + \frac{\partial f_a}{\partial q_c}dq_c + \frac{\partial f_a}{\partial q_d}dq_d \\ &= \frac{\partial f_a}{\partial q_a}[\frac{1}{4}(dq + dq^i + dq^j + dq^k)] \\ &+ \frac{\partial f_a}{\partial q_b}[\frac{1}{4i}(dq + dq^i - dq^j - dq^k)] \\ &+ \frac{\partial f_a}{\partial q_c}[\frac{1}{4j}(dq - dq^i + dq^j - dq^k)] \\ &+ \frac{\partial f_a}{\partial q_d}[\frac{1}{4k}(dq - dq^i - dq^j + dq^k)] \\ &= \frac{1}{4}(\frac{\partial f_a}{\partial q_a} - i\frac{\partial f_a}{\partial q_b} - j\frac{\partial f_a}{\partial q_c} - k\frac{\partial f_a}{\partial q_d})dq \\ &+ \frac{1}{4}(\frac{\partial f_a}{\partial q_a} - i\frac{\partial f_a}{\partial q_b} + j\frac{\partial f_a}{\partial q_c} + k\frac{\partial f_a}{\partial q_d})dq^i \\ &+ \frac{1}{4}(\frac{\partial f_a}{\partial q_a} + i\frac{\partial f_a}{\partial q_b} - j\frac{\partial f_a}{\partial q_c} + k\frac{\partial f_a}{\partial q_d})dq^j \\ &+ \frac{1}{4}(\frac{\partial f_a}{\partial q_a} + i\frac{\partial f_a}{\partial q_b} + j\frac{\partial f_a}{\partial q_c} - k\frac{\partial f_a}{\partial q_d})dq^k \end{aligned} \tag{6}$$

Similar expressions for idf_b, jdf_c, kdf_d can be derived in the same way. The sum of the four expressions give an expressions for df. Comparing the resulted expression with equation (5), we observe that the coefficient for dq should be the same, hence:

$$\begin{aligned} \frac{\partial f}{\partial q} &= \frac{1}{4}(\frac{\partial f_a}{\partial q_a} - i\frac{\partial f_a}{\partial q_b} - j\frac{\partial f_a}{\partial q_c} - k\frac{\partial f_a}{\partial q_d}) \\ &+ \frac{i}{4}(\frac{\partial f_b}{\partial q_a} - i\frac{\partial f_b}{\partial q_b} - j\frac{\partial f_b}{\partial q_c} - k\frac{\partial f_b}{\partial q_d}) \\ &+ \frac{j}{4}(\frac{\partial f_c}{\partial q_a} - i\frac{\partial f_c}{\partial q_b} - j\frac{\partial f_c}{\partial q_c} - k\frac{\partial f_c}{\partial q_d}) \\ &+ \frac{k}{4}(\frac{\partial f_d}{\partial q_a} - i\frac{\partial f_d}{\partial q_b} - j\frac{\partial f_d}{\partial q_c} - k\frac{\partial f_d}{\partial q_d}) \\ &= \frac{1}{4}(\frac{\partial f}{\partial q_a} - \frac{\partial f}{\partial q_b}i - \frac{\partial f}{\partial q_c}j - \frac{\partial f}{\partial q_d}k) \end{aligned} \tag{7}$$

Therefore, $\dfrac{\partial f(q)}{\partial q}$ is given by

$$\frac{\partial f(q)}{\partial q} = \frac{1}{4}(\frac{\partial f(q)}{\partial q_a} - \frac{\partial f(q)}{\partial q_b}i - \frac{\partial f(q)}{\partial q_c}j - \frac{\partial f(q)}{\partial q_d}k) \tag{8}$$

Expressions for $\partial f/\partial q^i$, $\partial f/\partial q^j$ and $\partial f/\partial q^k$ can be derived similarly. Note that, in general $\partial f(q)/\partial q_b$ is a quaternion, therefore $\partial f(q)/\partial q_b i \neq i\partial f(q)/\partial q_b$, i.e., the two factors do not commute. The same argument applies to the last two terms in equation (8).

$f(q)$ can also be viewed as a function of q^* and its involutions. Following the same arguments, we can also find the derivative of $f(q)$ with respect to q^*, which is given by

$$\frac{\partial f(q)}{\partial q^*} = \frac{1}{4}(\frac{\partial f(q)}{\partial q_a} + \frac{\partial f(q)}{\partial q_b}i + \frac{\partial f(q)}{\partial q_c}j + \frac{\partial f(q)}{\partial q_d}k) \tag{9}$$

where $q^* = q_a - q_b i - q_c j - q_d k$.

With these results, we can then calculate the derivatives of some simple quaternion functions. For example, we easily obtain

$$\frac{\partial q}{\partial q} = 1,\ \frac{\partial q}{\partial q^*} = -\frac{1}{2}\ . \tag{10}$$

On the other hand, the product rule is not true in general due to the non-commutativity of quaternion products. However, it holds for the differentiation of quaternion-valued functions to real variables. Suppose $f(q)$ and $g(q)$ are two quaternion-valued functions of the quaternion variable q, and q_a is the real variable. Then we can have the following result

$$\begin{aligned} \frac{\partial f(q)g(q)}{\partial q_a} &= \frac{\partial}{\partial q_a}(f_a + if_b + jf_c + kf_d)g \\ &= \frac{\partial f_a g}{\partial q_a} + i\frac{\partial f_b g}{\partial q_a} + j\frac{\partial f_c g}{\partial q_a} + k\frac{\partial f_d g}{\partial q_a} \\ &= (f_a\frac{\partial g}{\partial q_a} + \frac{\partial f_a}{\partial q_a}g) + i(f_b\frac{\partial g}{\partial q_a} + \frac{\partial f_b}{\partial q_a}g) \\ &\quad + j(f_c\frac{\partial g}{\partial q_a} + \frac{\partial f_c}{\partial q_a}g) + k(f_d\frac{\partial g}{\partial q_a} + \frac{\partial f_d}{\partial q_a}g) \\ &= (f_a + if_b + jf_c + kf_d)\frac{\partial g}{\partial q_a} \\ &\quad + (\frac{\partial f_a}{\partial q_a} + i\frac{\partial f_b}{\partial q_a} + j\frac{\partial f_c}{\partial q_a} + k\frac{\partial f_d}{\partial q_a})g \\ &= f(q)\frac{\partial g(q)}{\partial q_a} + \frac{\partial f(q)}{\partial q_a}g(q) \end{aligned} \tag{11}$$

When the quaternion variable q is replaced by a quaternion-valued vector $\mathbf{w}$, given by

$$\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_M]^T \tag{12}$$

where $w_m = a_m + b_m i + c_m j + d_m k$, $m = 1, ..., M$, the differentiation of $f(\mathbf{w})$ with respect to $\mathbf{w}$ can be derived using a combination of (8) as follows

$$\frac{\partial f}{\partial \mathbf{w}} = \frac{1}{4} \begin{bmatrix} \frac{\partial f}{\partial a_1} - \frac{\partial f}{\partial b_1} i - \frac{\partial f}{\partial c_1} j - \frac{\partial f}{\partial d_1} k \\ \frac{\partial f}{\partial a_2} - i \frac{\partial f}{\partial b_2} i - \frac{\partial f}{\partial c_2} j - \frac{\partial f}{\partial d_2} k \\ \vdots \\ \frac{\partial f}{\partial a_M} - \frac{\partial f}{\partial b_M} i - \frac{\partial f}{\partial c_M} j - \frac{\partial f}{\partial d_M} k \end{bmatrix} \tag{13}$$

Similarly, we define $\dfrac{\partial f}{\partial \mathbf{w}^*}$ as

$$\frac{\partial f}{\partial \mathbf{w}^*} = \frac{1}{4} \begin{bmatrix} \frac{\partial f}{\partial a_1} + \frac{\partial f}{\partial b_1} i + \frac{\partial f}{\partial c_1} j + \frac{\partial f}{\partial d_1} k \\ \frac{\partial f}{\partial a_2} + \frac{\partial f}{\partial b_2} i + \frac{\partial f}{\partial c_2} j + \frac{\partial f}{\partial d_2} k \\ \vdots \\ \frac{\partial f}{\partial a_M} + \frac{\partial f}{\partial b_M} i + \frac{\partial f}{\partial c_M} j + \frac{\partial f}{\partial d_M} k \end{bmatrix} \tag{14}$$

Obviously, when $M = 1$, (13) and (14) are reduced to (8) and (9), respectively.

B. The QLMS algorithm

The output $y[n]$ and error $e[n]$ of a standard adaptive filter can be expressed as

$$y[n] = \mathbf{w}^T[n]\mathbf{x}[n] \tag{15}$$

$$e[n] = d[n] - \mathbf{w}^T[n]\mathbf{x}[n], \tag{16}$$

where $\mathbf{w}[n]$ is the adaptive weight vector with a length of M, $d[n]$ is the reference signal, $\mathbf{x}[n] = [x[n-1], x[n-2], \cdots, x[n-M]]^T$ is the input sample sequence, and $\{\cdot\}^T$ denotes the transpose operation. The cost function with the quaternion-valued error is $J_0[n] = e[n]e^*[n]$. Its gradient is given by

$$\nabla_{\mathbf{w}^*} J_0[n] = \frac{\partial J_0[n]}{\partial \mathbf{w}^*} \tag{17}$$

$$\nabla_{\mathbf{w}} J_0[n] = \frac{\partial J_0[n]}{\partial \mathbf{w}} \tag{18}$$

with respect to $\mathbf{w}^*[n]$ and $\mathbf{w}[n]$, respectively. According to [10], [11], the conjugate gradient gives the maximum steepness direction for the optimization surface. Therefore, the conjugate gradient $\nabla_{\mathbf{w}^*} J_0[n]$ will be used to derive the update of the coefficient weight vector.

First we have

$$J_0[n] = d[n]d^*[n] - d[n]\mathbf{x}^H[n]\mathbf{w}^*[n] - \mathbf{w}^T[n]\mathbf{x}[n]d^*[n] + \mathbf{w}^T[n]\mathbf{x}[n]\mathbf{x}^H[n]\mathbf{w}^*[n] \tag{19}$$

For different parts, we obtain the following results

$$\frac{\partial (d[n]d^*[n])}{\partial \mathbf{w}^*[n]} = 0 \tag{20}$$

$$\frac{\partial (d[n]\mathbf{x}^H[n]\mathbf{w}^*[n])}{\partial \mathbf{w}^*[n]} = d[n]\mathbf{x}^*[n] \tag{21}$$

$$\frac{\partial (\mathbf{w}^T[n]\mathbf{x}[n]d^*[n])}{\partial \mathbf{w}^*[n]} = -\frac{1}{2} d[n]\mathbf{x}^*[n] \tag{22}$$

$$\frac{\partial (\mathbf{w}^T[n]\mathbf{x}[n]\mathbf{x}^H[n]\mathbf{w}^*[n])}{\partial \mathbf{w}^*[n]} = \frac{1}{2}\mathbf{w}^T[n]\mathbf{x}[n]\mathbf{x}^*[n] \tag{23}$$

Then we have the final gradient result

$$\nabla_{\mathbf{w}^*} J_0[n] = -\frac{1}{2} e[n]\mathbf{x}^*[n]. \tag{24}$$

With the general update equation for the weight vector

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \mu \nabla_{\mathbf{w}^*} J_0[n], \tag{25}$$

we arrive at the following update equation for the QLMS algorithm with step size μ

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu(e[n]\mathbf{x}^*[n]). \tag{26}$$

C. The AQLMS algorithm

Recently, to fully exploit the second-order statistics of the signals, an augmented formulation of the data vector has been proposed, first for complex-valued signals and then for quaternion-valued ones. For complex-valued signals, the augmented vector is composed of the original data and its conjugate, while for the latter, due to existence of the three perpendicular quaternion involutions, the choice for the augmented vector is not unique. Without loss of generality, here we adopt the simplest formulation by combining the data vector $\mathbf{x}[n]$ and its conjugate $\mathbf{x}^*[n]$ to produce an augmented vector $\mathbf{x}_a[n] = \left[\mathbf{x}^T[n] \ \ \mathbf{x}^H[n]\right]^T$ [12], where $\{\cdot\}^H$ is a combination of the operations of $\{\cdot\}^T$ and $\{\cdot\}^*$ for a quaternion. For such a "widely linear" model, the quaternion-valued output for the conjugate part of the input is given by

$$\hat{y}[n] = \mathbf{g}^T[n]\mathbf{x}^*[n], \tag{27}$$

where $\mathbf{g}[n]$ denotes the weight vector for the conjugate part of the input $\mathbf{x}[n]$.

As to the AQLMS algorithm, the update of the weight vector of the conjugate part $\mathbf{g}[n]$ can be found with the same method as that of the QLMS in (26), i.e.

$$\mathbf{g}[n+1] = \mathbf{g}[n] + \mu(e[n]\mathbf{x}[n]). \tag{28}$$

With the augmented weight vector $\mathbf{h}_a[n]$ defined as

$$\mathbf{h}_a[n] = \left[\mathbf{w}^T[n] \ \ \mathbf{g}^T[n]\right]^T, \tag{29}$$

we obtain the following update equation

$$\mathbf{h}_a[n+1] = \mathbf{h}_a[n] + \mu(e_a[n]{\mathbf{x}_a}^*[n]) \tag{30}$$

where $e_a[n] = d[n] - {\mathbf{h}_a}^T[n]\mathbf{x}_a[n]$.

III. Application to CFD data

CFD is a branch of fluid mechanics. It uses numerical approaches to solve fluid flow problems.

A. Fluid Dynamics Equations

The Navier-Stokes equations are the basis of fluid problems. They are essentially the mathematical formulation of the Newton's second law applied to fluid motions. The general expression of the equations is

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}) = -\nabla P + \eta \Delta \mathbf{u} \tag{31}$$

where $\mathbf{u}$ is the fluid velocity at a particular spatial location at a given time, P is the pressure and ρ is the fluid density. The left hand side of the equation is the acceleration of the fluid, whilst on the right side are (the gradient of) the forces, including pressure and viscous force. Together with the conservation of mass and suitable boundary conditions, the Navier-Stokes equations can model a large class of fluid motions accurately [13].

B. Turbulence

The second term on the left hand side of equation (31) represents the contribution from the advection of fluid particles to fluid acceleration, and is customarily called the inertial force. The second term on the right hand side represents the viscous force. The ratio of these two forces is defined as the Reynolds number (Re). As it turns out, when Re is large, the flows tend to become unstable and generate a spectrum of high frequency components in the velocity signal. Such a regime of fluid motions is called turbulence. Atmospheric flows, including the wind fields around wind farms, are always turbulence [14]. Due to the presence of the high frequency components, the CFD calculation of the velocity signal in turbulent wind fields becomes very time consuming unless simplifying models are introduced.

C. Direct numerical simulation (DNS)

DNS solves the Navier-Stokes equations directly without any turbulence models. The advantage of this method is that it is simple as well as accurate with complete information. However, the computational cost can be very high if Re is large. Therefore, this method is not yet applicable to practical situations, for example, the atmospheric flows we will deal with [13]. Nevertheless, as a first step, we choose to use DNS to generate the velocity signals in this study.

D. Data Generation Using CFD

The velocity signals are generated by DNS, where the Navier-Stokes equations are solved using a pseudo-spectral method. The CFD code is written in FORTRAN 90. Running the code, we generate a time series of three dimensional turbulent wind velocity fields in a 3-D periodic box. We consider the flow field as an idealized wind field with the mean velocity having been subtracted, and the signal normalized.

IV. Application to Adaptive Beamforming

A. Quaternionic array signal model

A uniform linear array (ULA) with M crossed-dipole pairs is shown in Fig. 1. These pairs are located along the y-axis with an adjacent spacing d, and at each location the two crossed components are parallel to x-axis and y-axis, respectively. Assume there is a far-field incident signal with direction of arrival (DOA) defined by the angles θ and ϕ impinging upon the array from the y-z plane, so that $\phi = \pi/2$ as well as $\phi = -\pi/2$, and $0 \leq \theta \leq \pi/2$. As a result, the spatial steering vector for the signal is expressed as

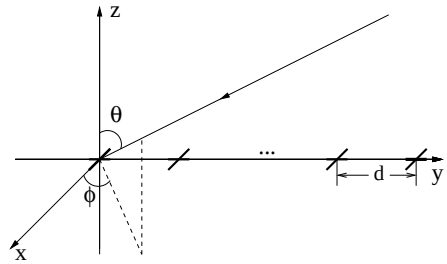

Fig. 1. A ULA with crossed-dipoles.

$$\begin{aligned} \mathbf{S}_c(\theta, \phi) &= [1, e^{-j2\pi d sin\theta sin\phi/\lambda}, \\ &\quad \cdots, e^{-j2\pi(M-1)d sin\theta sin\phi/\lambda}]^T \end{aligned} \tag{32}$$

where λ is the wavelength of the incident signal. For a crossed dipole the spatial-polarization coherent vector can be given by [15], [16]

$$\mathbf{S}_p(\theta, \phi, \gamma, \eta) = \begin{cases} [-cos\gamma, cos\theta sin\gamma e^{j\eta}] & \text{for } \phi = \pi/2 \\ [cos\gamma, -cos\theta sin\gamma e^{j\eta}] & \text{for } \phi = -\pi/2 \end{cases} \tag{33}$$

where γ is the auxiliary polarization angle with $\gamma \in [0, \pi/2]$, and the $\eta \in [-\pi, \pi]$ is the polarization phase difference.

The array structure can be divided into two sub-arrays. One is parallel to the x-axis and the other is parallel to the y-axis. The complex-valued steering vector of the x-axis sub-array is given by

$$\mathbf{S}_x(\theta, \phi, \gamma, \eta) = \begin{cases} -cos\gamma \mathbf{S}_c(\theta, \phi) & \text{for } \phi = \pi/2 \\ cos\gamma \mathbf{S}_c(\theta, \phi) & \text{for } \phi = -\pi/2 \end{cases} \tag{34}$$

and for the y-axis it is expressed as

$$\mathbf{S}_y(\theta, \phi, \gamma, \eta) = \begin{cases} cos\theta sin\gamma e^{j\eta} \mathbf{S}_c(\theta, \phi) & \text{for } \phi = \pi/2 \\ -cos\theta sin\gamma e^{j\eta} \mathbf{S}_c(\theta, \phi) & \text{for } \phi = -\pi/2 \end{cases} \tag{35}$$

Combining these two steering vectors together, we have a quaternion-valued composite steering vector given as below

$$\mathbf{S}_q(\theta, \phi, \gamma, \eta) = \mathbf{S}_x(\theta, \phi, \gamma, \eta) + i\mathbf{S}_y(\theta, \phi, \gamma, \eta). \tag{36}$$

The response of the array is

$$r(\theta, \phi, \gamma, \eta) = \mathbf{w}^H \mathbf{S}_q(\theta, \phi, \gamma, \eta) \tag{37}$$

where $\mathbf{w}$ is the quaternion-valued weight vector.

B. Reference signal based adaptive beamforming

When a reference signal $d[n]$ is available, adaptive beamforming can be implemented by the standard adaptive filter structure, as shown in Fig. 2, where $x_m[n]$, $m = 1, 2, \cdots, M$ are the received quaternion-valued vector sensor signals,

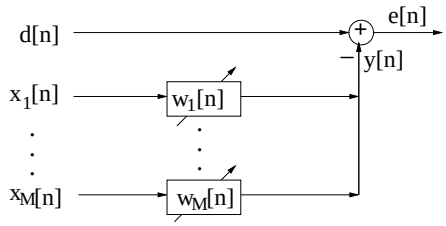

Fig. 2. Structure of a reference signal based adaptive beamformer.

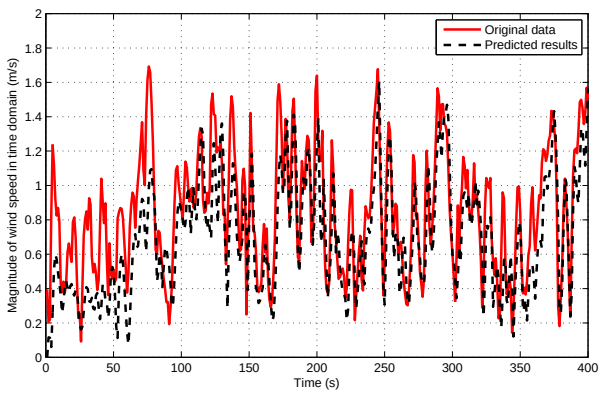

Fig. 3. Prediction results using the QLMS algorithm.

$w_m[n]$, $m = 1, 2, \cdots, M$ are the corresponding quaternion-valued coefficients, $y[n]$ is the beamformer output and $e[n]$ is the error signal.

V. Simulation Results

A. Scenario one

In this part, both the QLMS and the AQLMS algorithms are applied to the wind data generated by CFD simulations with a sampling frequency of 1 Hz. The parameters are as follows. The step size is $\mu = 2.5 \times 10^{-4}$ and the adaptive filter length is $L = 16$. The prediction step is 2. The adaptive weight vector is initialized as an all-zero vector. Fig. 3 and Fig. 4 show the results for the QLMS and AQLMS algorithms, respectively.

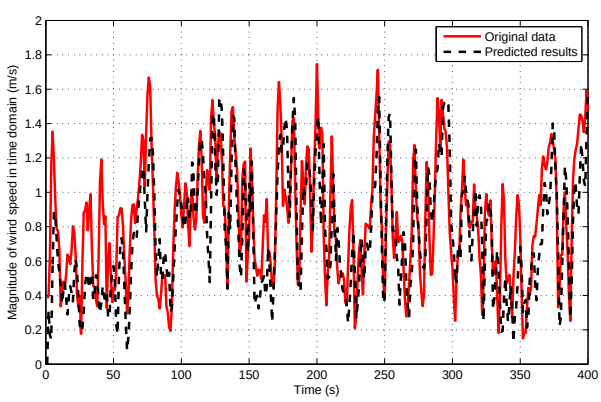

Fig. 4. Prediction results using the AQLMS algorithm.

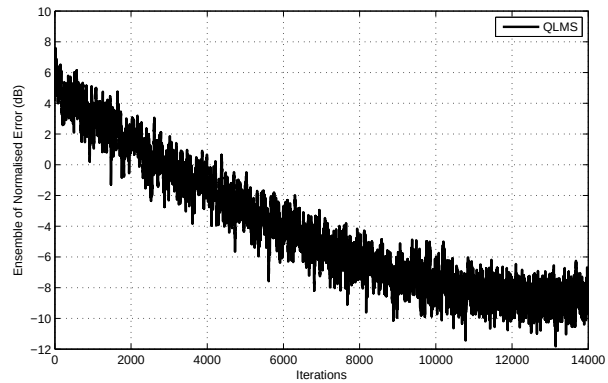

Fig. 5. Learning curve using the QLMS algorithm for adaptive beamforming.

As we can see from the results, both algorithms can track the change of the wind speed signal effectively.

B. Scenario two

Now we run simulations for the adaptive beamforming scenario. The vector sensor array with 10 crossed-dipoles and half-wavelength spacing is considered to obtain the output using the QLMS algorithm. The stepsize μ here is set to be 1×10^{-6}. A desired signal with 20 dB SNR impinges from the broadside and two interfering signals with the signal to interference ratio (SIR) of 0 dB arrive from $30°$ and $-20°$, respectively. All the signals have the same polarisation of $(\gamma, \eta) = (0, 0)$. The learning curve averaged over 100 simulation runs is shown in Fig. 5, where we can see the normalised error has reached about -10 dB, indicating an effective beamforming operation.

VI. Conclusion

In this paper, a general quaternion-valued gradient operator has been derived in detail, based on which two adaptive algorithms were developed including the QLMS and the AQLMS algorithms. These algorithms were applied to two different areas. One is to combine with the classic computational fluid dynamics (CFD) approach in wind profile prediction and the other one is to apply the result to the adaptive beamforming problem for vector sensor arrays. Simulation results have shown that the derived algorithms can work in different scenarios effectively, highlighting the importance and usefulness of the derived gradient operator. One important note is that although there have been some derivations of this operator in literature with different level of details, this is the first time to give the most general form with a solid theoretical basis.

VII. Acknowledgements

This work is partially funded by National Grid UK.

REFERENCES

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

[2] X. R. Zhang, W. Liu, Y. G. Xu, and Z. W. Liu, "Quaternion-valued robust adaptive beamformer for electromagnetic vector-sensor arrays with worst-case constraint," *Signal Processing*, vol. 104, pp. 274–283, November 2014.

[3] M. B. Hawes and W. Liu, "A quaternion-valued reweighted minimisation approach to sparse vector sensor array design," in *Proc. of the International Conference on Digital Signal Processing*, Hong Kong, August 2014.

[4] C. C. 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] M. D. Jiang, W. Liu, Y. Li, and X. R. Zhang, "Frequency-domain quaternion-valued adaptive filtering and its application to wind profile prediction," in *Proc. of the IEEE TENCON Conference*, Xi'an, China, October 2013.

[6] H. L. Van Trees, *Optimum Array Processing, Part IV of Detection, Estimation, and Modulation Theory*. New York: Wiley, 2002.

[7] W. Liu and S. Weiss, *Wideband Beamforming: Concepts and Techniques*. Chichester, UK: John Wiley & Sons, 2010.

[8] A. Nehorai, K. C. Ho, and B. T. G. Tan, "Minimum-noise-variance beamformer with an electromagnetic vector sensor," *IEEE Transactions on Signal Processing*, vol. 47, no. 3, pp. 601–618, March 1999.

[9] T. A. Ell and S. J. Sangwine, "Quaternion involutions and anti-involutions," *Computers & Mathematics with Applications*, vol. 53, no. 1, pp. 137–143, 2007.

[10] D. P. Mandic, C. Jahanchahi, and C. C. Took, "A quaternion gradient operator and its applications," *IEEE Signal Processing Letters*, vol. 18, no. 1, pp. 47–50, 2011.

[11] D. Brandwood, "A complex gradient operator and its application in adaptive array theory," in *IEE Proceedings H (Microwaves, Optics and Antennas)*, vol. 130, no. 1. IET, 1983, pp. 11–16.

[12] C. C. Took and D. P. Mandic, "A quaternion widely linear adaptive filter," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4427–4431, 2010.

[13] J. Ferzige and M. Peric, *Computational methods for fluid dynamics*. Springer, 2001.

[14] S. B. Pope, *Turbulent flows*. Cambridge university press, 2000.

[15] R. Compton Jr, "On the performance of a polarization sensitive adaptive array," *IEEE Transactions on Antennas and Propagation*, vol. 29, no. 5, pp. 718–725, 1981.

[16] J. Li and R. Compton Jr, "Angle and polarization estimation using esprit with a polarization sensitive array," *IEEE Transactions on Antennas and Propagation*, vol. 39, pp. 1376–1383, 1991.