Proceedings Paper:

Gou, X., Liu, Z., Liu, W. et al. (1 more author) (2015) Three-Dimensional Wind Profile Prediction with Trinion-Valued Adaptive Algorithms. In: 2015 IEEE International Conference on Digital Signal Processing (DSP). , 21-24 July 2015, Singapore. IEEE , pp. 566-569.

https://doi.org/10.1109/ICDSP.2015.7251937

Three-Dimensional Wind Profile Prediction with Trinion-Valued Adaptive Algorithms

Xiaoming Gou*†, Zhiwen Liu*, Wei Liu† and Yougen Xu*
*School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China
†Communications Research Group, Department of Electronic and Electrical Engineering,
University of Sheffield, Sheffield S1 3JD, United Kingdom
Email: w.liu@sheffield.ac.uk

***Abstract*—The problem of three-dimensional (3-D) wind profile prediction is addressed based a trinion wind model, which inherently reckons the coupling of the three perpendicular components of a wind field. The augmented trinion statistics are developed and employed to enhance the prediction performance due to its full exploitation of the second-order statistics. The proposed trinion domain processing can be regarded as a more compact version of the existing quaternion-valued approach, with a lower computational complexity. Simulations based on recorded wind data are provided to demonstrate the effectiveness of the proposed methods.**

***Index Terms*—wind profile prediction, trinion-valued representation, least mean squares, adaptive filtering, widely linear processing**

I. Introduction

Hypercomlex numbers are high-dimensional extensions of real numbers [1], [2], and they have been introduced to solve multivariate signal processing problems, such as colour image processing [3]–[6], vector-sensor array signal processing [7]–[12], human gesture spotting [13], [14], wind profile prediction [15]–[17], [22], and wireless communications [23]. In the last application, anemometer readings are modeled with hypercomplex numbers and the wind profile is predicted by adaptive filtering algorithms. In particular, pure quaternions have been widely used to model three-dimensional (3D) wind speed. When external atmospheric parameters are available, a full quaternion-valued model can be considered [15].

The quaternion-valued model leads to improved performance over real-valued models, since it accounts for the coupling of the wind measurements and can be extended to exploiting the augmented quaternion statistics [15]. However, pure quaternions do not form a mathematical ring [18], since the product of two pure quaternions is not a pure quaternion in general. As a result, the related adaptive algorithms for 3-D wind profile prediction initialised in the pure quaternion will have to work in the full quaternion domain. The prediction results have to be truncated from full quaternions to pure quaternions, implying redundant computations in the update process. For example, it takes 16 real-valued multiplications and 12 real-valued additions to implement a full-quaternion-valued multiplication. By comparison, it only takes 9 real-valued multiplications and 6 real-valued additions to calculate the multiplication of any two 3-D numbers.

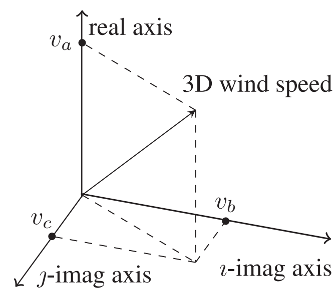

Fig. 1. Trinion-valued three-dimensional wind speed model.

In this work, we aim to develop a more compact 3-D wind speed model based on a 3-D mathematical ring called trinions, termed by an anonymous author who provided all possible definitions of 3-D numbers [19]. We first define the gradient operation in the trinion domain and then derive the least mean squares (LMS) adaptive algorithm, which is further extended to augmented statistics.

The rest of this paper is organised as follows. A brief introduction to trinions is provided in Section II. The trinion-valued LMS (TLMS) algorithm is developed in Section III, and the augmented TLMS (ATLMS) algorithm in Section IV. Simulation results are presented in Section V and conclusions are drawn in Section VI.

II. Trinions

As shown in Fig. 1, the 3-D wind speed is a tri-variate signal composed of three perpendicular components, and can be modeled with 3-D hypercomplex numbers (see Fig. 1). There are various definitions of a 3-D number v composed of one real part (v_a) and two imaginary parts (v_b and v_c),

$$v = v_a + \imath v_b + \jmath v_c, \tag{1}$$

according to definitions about the relationships among the three base elements 1, $\imath$, and $\jmath$.

To form an Abelian group of these three, the following rules apply [20]:

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

Trinions subject to the above rules form a commutative ring, namely, for two trinions v and w, we have $vw = wv$.

The modulus of v is given by [20]

$$|v| = \sqrt{v_a^2 + v_b^2 + v_c^2}. \tag{3}$$

We define the following conjugate of v,

$$v^* = v_a - \jmath v_b - \imath v_c, \tag{4}$$

so that $|v|^2$ is equal to the real part of vv^*, i.e. $|v|^2 = \Re(vv^*)$.

To develop an LMS-like algorithm in the trinion domain, the trinion-valued gradient needs to be defined. In the complex domain, a variable z and its conjugate can be viewed as two independent variables, based on which the complex-valued gradient can be defined [21]. Similarly, in the quaternion domain, a variable q and its three involutions can be viewed as four independent variables, based on which the quaternion-valued gradient can be derived [22], [24]. However, to our best knowledge, the trinion involution is not available in general. In this paper, we define the following gradients of a function $f(\boldsymbol{x})$ with respect to the trinion-valued variable $\boldsymbol{x}$ and its conjugate,

$$\begin{aligned} \nabla_{\boldsymbol{x}} f &= \frac{1}{3}\big(\nabla_{\boldsymbol{x}_a} f - \jmath \nabla_{\boldsymbol{x}_b} f - \imath \nabla_{\boldsymbol{x}_c} f\big), \\ \nabla_{\boldsymbol{x}^*} f &= \frac{1}{3}\big(\nabla_{\boldsymbol{x}_a} f + \imath \nabla_{\boldsymbol{x}_b} f + \jmath \nabla_{\boldsymbol{x}_c} f\big), \end{aligned} \tag{5}$$

respectively, where $\boldsymbol{x} = \boldsymbol{x}_a + \imath \boldsymbol{x}_b + \jmath \boldsymbol{x}_c$.

Unlike the quaternion-valued gradient, the trinion-valued gradient remains the same, no matter which side of the sub-gradients the imaginary units $\imath$ and $\jmath$ are on, since trinions are commutative. We can then calculate the derivatives of some simple functions, for instance,

$$\begin{aligned} \frac{\partial x}{\partial x} &= \frac{\partial x^*}{\partial x^*} = 1, \\ \frac{\partial x}{\partial x^*} &= \frac{\partial x^*}{\partial x} = \frac{1 - \imath + \jmath}{3}. \end{aligned} \tag{6}$$

III. Trinion-valued LMS Filtering

Similar to the LMS algorithm in the complex-valued domain, the trinion-valued error $e(n)$ is given by

$$e(n) = d(n) - \boldsymbol{w}^{\mathrm{T}}(n)\boldsymbol{x}(n), \tag{7}$$

where $d(n)$ is the reference signal, $\boldsymbol{w}(n)$ is the weight vector, $\boldsymbol{x}(n) = [x(n-P); x(n-P-1); \cdots ; x(n-L-P+1)]^T$ is the filter input, L is the filter length, and P is the prediction step. The cost function is expressed as

$$J(n) = |e(n)|^2. \tag{8}$$

Using the steepest descent method, the following gradient need to be calculated

$$\nabla_{\boldsymbol{w}^*} J(n) = \frac{1}{3}\big[\nabla_{\boldsymbol{w}_a} J(n) + \imath \nabla_{\boldsymbol{w}_b} J(n) + \jmath \nabla_{\boldsymbol{w}_c} J(n)\big]. \tag{9}$$

For details of the calculation, please refer to Appendix at the end. Using results there, the update of the weight vector can be obtained as

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu e(n)\boldsymbol{x}^*(n), \tag{10}$$

where μ is the step size. The algorithm formulated above is termed as the trinion-valued LMS (TLMS) algorithm.

IV. Augmented Trinion Statistics

A zero-mean trinion-valued vector $\boldsymbol{v}$ is composed of three zero-mean real-valued random vectors $\boldsymbol{v}_a$, $\boldsymbol{v}_b$, and $\boldsymbol{v}_c$, and their complete second-order statistics can be found in the following six real-valued covariance matrices:

$$\begin{aligned} \boldsymbol{C}_{\boldsymbol{v}_a\boldsymbol{v}_a} &= E\{\boldsymbol{v}_a\boldsymbol{v}_a^{\mathrm{T}}\}, \quad \boldsymbol{C}_{\boldsymbol{v}_b\boldsymbol{v}_b} = E\{\boldsymbol{v}_b\boldsymbol{v}_b^{\mathrm{T}}\}, \\ \boldsymbol{C}_{\boldsymbol{v}_c\boldsymbol{v}_c} &= E\{\boldsymbol{v}_c\boldsymbol{v}_c^{\mathrm{T}}\}, \quad \boldsymbol{C}_{\boldsymbol{v}_a\boldsymbol{v}_b} = E\{\boldsymbol{v}_a\boldsymbol{v}_b^{\mathrm{T}}\}, \\ \boldsymbol{C}_{\boldsymbol{v}_b\boldsymbol{v}_c} &= E\{\boldsymbol{v}_b\boldsymbol{v}_c^{\mathrm{T}}\}, \quad \boldsymbol{C}_{\boldsymbol{v}_c\boldsymbol{v}_a} = E\{\boldsymbol{v}_c\boldsymbol{v}_a^{\mathrm{T}}\}. \end{aligned} \tag{11}$$

And they can be more efficiently represented by three trinion-valued covariance matrices:

$$\begin{aligned} \boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}} &= E\{\boldsymbol{v}\boldsymbol{v}^{\mathrm{H}}\}, \\ \boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\imath}} &= E\{\boldsymbol{v}\boldsymbol{v}^{\imath\mathrm{H}}\}, \\ \boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\jmath}} &= E\{\boldsymbol{v}\boldsymbol{v}^{\jmath\mathrm{H}}\}, \end{aligned} \tag{12}$$

where superscript $^{\mathrm{H}}$ denotes the Hermitian transpose and the two additional mappings of $\boldsymbol{v}$ are defined as

$$\begin{aligned} \boldsymbol{v}^{\imath} &= \boldsymbol{v}_b - \imath \boldsymbol{v}_a - \jmath \boldsymbol{v}_c, \\ \boldsymbol{v}^{\jmath} &= \boldsymbol{v}_c - \imath \boldsymbol{v}_b - \jmath \boldsymbol{v}_a. \end{aligned} \tag{13}$$

Neither of these two mappings is an involution, and they are defined as shorthand notations only. Then we can obtain

$$\begin{aligned} \boldsymbol{C}_{\boldsymbol{v}_a\boldsymbol{v}_a} &= \frac{1}{2}\Re(\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}} + \jmath\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\imath}}), \\ \boldsymbol{C}_{\boldsymbol{v}_b\boldsymbol{v}_b} &= \frac{1}{2}\Re(\jmath\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\imath}} - \jmath\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\imath}}), \\ \boldsymbol{C}_{\boldsymbol{v}_c\boldsymbol{v}_c} &= \frac{1}{2}\Re(\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}} - \imath\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\jmath}}), \\ \boldsymbol{C}_{\boldsymbol{v}_a\boldsymbol{v}_b} &= \frac{1}{2}\Re(\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\imath}} + \jmath\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\jmath}}), \\ \boldsymbol{C}_{\boldsymbol{v}_b\boldsymbol{v}_c} &= \frac{1}{2}\Re(\imath\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}} - \jmath\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\jmath}}), \\ \boldsymbol{C}_{\boldsymbol{v}_c\boldsymbol{v}_a} &= \frac{1}{2}\Re(\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\imath}} - \imath\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}}). \end{aligned} \tag{14}$$

To take the complete second-order statistics into consideration, we need to make use of the augmented input vector composed of the original input vector $\boldsymbol{x}(n)$ and its two mappings $\boldsymbol{x}^{\imath}(n), \boldsymbol{x}^{\jmath}(n)$, namely,

$$\boldsymbol{x}^{\mathrm{aug}}(n) = \begin{bmatrix} \boldsymbol{x}(n) \\ \boldsymbol{x}^{\imath}(n) \\ \boldsymbol{x}^{\jmath}(n) \end{bmatrix}, \tag{15}$$

and the predicted estimate $y(n)$ will be given by

$$\begin{aligned} y(n) &= \boldsymbol{w}^{\mathrm{augT}}(n)\boldsymbol{x}^{\mathrm{aug}}(n) \\ &= \boldsymbol{w}_1^{\mathrm{T}}(n)\boldsymbol{x}(n) + \boldsymbol{w}_2^{\mathrm{T}}(n)\boldsymbol{x}^{\imath}(n) + \boldsymbol{w}_3^{\mathrm{T}}(n)\boldsymbol{x}^{\jmath}(n), \end{aligned} \tag{16}$$

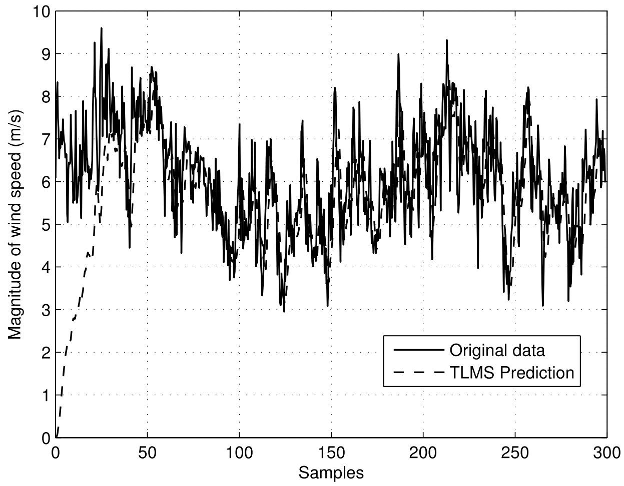

Fig. 2. Predicted result from the TLMS algorithm.

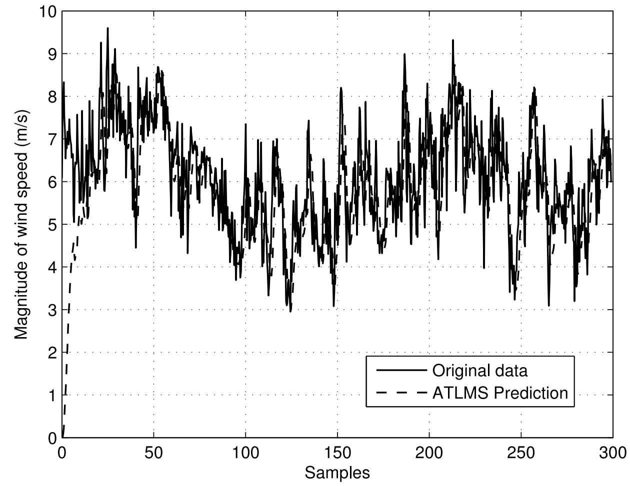

Fig. 3. Predicted result from the ATLMS algorithm.

where $\boldsymbol{w}^{\mathrm{aug}} = [\boldsymbol{w}_1; \boldsymbol{w}_2; \boldsymbol{w}_3]$. Analogous to the derivation of the TLMS algorithm, the update equation of the augmented TLMS (ATLMS) algorithm is given by

$$\boldsymbol{w}^{\mathrm{aug}}(n+1) = \boldsymbol{w}^{\mathrm{aug}}(n) + \rho e^{\mathrm{aug}}(n)\boldsymbol{x}^{\mathrm{aug}*}(n), \tag{17}$$

where ρ is the step size.

The computational complexity for each update of the weight vector of the trinion-based and quaternion-based filtering algorithms are summarised in Table I. It can be seen that the trinion model can effectively reduce the computational complexity compared to the quaternion model.

TABLE I
COMPUTIONAL COMPLEXITY PER UPDATE OF THE WEIGHT VECTOR

Algorithm	Real multiplications	Real additions
QLMS [17]	$16L+4$	$16L$
Augmented QLMS	$64L+4$	$64L$
TLMS	$9L+3$	$9L$
Augmented TLMS	$27L+3$	$27L$

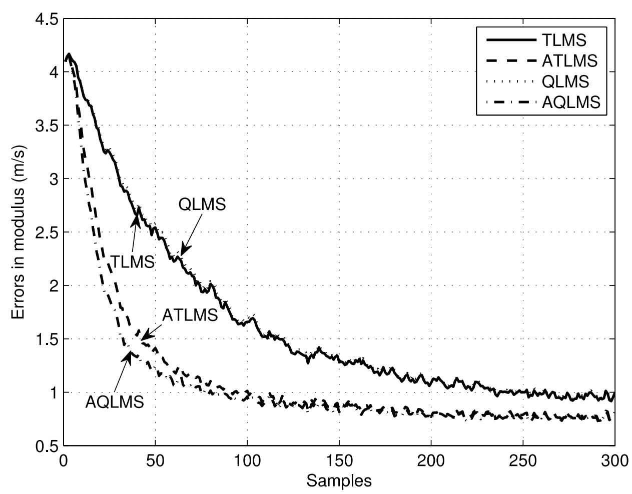

Fig. 4. Averaged learning curves.

V. SIMULATIONS

In this section, both proposed algorithms (TLMS and ATLMS) are applied to anemometer readings provided by Google's RE<C Initiative [25]. The wind speed measured on May 25, 2011 is used for demonstration. The step size is set to be 6×10^{-5}. The filter length is 8, and the prediction step is 1. All algorithms are initialised with zeros. The predicted results provided by TLMS and ATLMS algorithms are shown in Figs. 2 and 3, respectively. From the results, we can see that both algorithms can track the wind data effectively.

The learning curves averaged over 200 trials of the proposed algorithms are shown in Fig. 4, compared with the quaternion-based QLMS and AQLMS algorithms. It can be observed that both augmented algorithms (AQLMS and ATLMS) have a faster convergence rate than the original ones (QLMS and TLMS), due to their full exploitation of the second-order statistics. Meanwhile, the proposed TLMS algorithm has a similar performance with the QLMS algorithm, while the ATLMS algorithm is comparable with the AQLMS algorithm. However, as shown in Table I, the proposed trinion-based algorithms have a much lower computational complexity.

VI. CONCLUSION

A compact model for three-dimensional wind profile prediction based on trinion algebra has been proposed, with two LMS-type adaptive filtering algorithms derived using partial and full trinion-valued second-order statistics, respectively. Numerical simulations using recorded wind data have shown that the two algorithms have a similar performance to their quaternion-valued counterparts, but with a much lower computational complexity.

APPENDIX

The cost function $J(n)$ as a function of real-valued variables is given by

$$\begin{aligned} J = & (d_a - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_a + \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_c + \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_b)^2 \\ & + (d_b - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_b - \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_a + \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_c)^2 \\ & + (d_c - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_c - \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_b - \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_a)^2, \end{aligned} \tag{18}$$

where the time index 'n' has been dropped for the sake of compact notation. Then the three component-wise gradients can be computed as

$$\begin{aligned} \nabla_{\boldsymbol{w}_a} J = 2\big[& (\boldsymbol{x}_a\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_b^{\mathrm{T}} + \boldsymbol{x}_c\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_a \\ & + (\boldsymbol{x}_b\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_c\boldsymbol{x}_b^{\mathrm{T}} - \boldsymbol{x}_a\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_b \\ & + (\boldsymbol{x}_c\boldsymbol{x}_a^{\mathrm{T}} - \boldsymbol{x}_a\boldsymbol{x}_b^{\mathrm{T}} - \boldsymbol{x}_b\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_c \\ & - (d_a\boldsymbol{x}_a + d_b\boldsymbol{x}_b + d_c\boldsymbol{x}_c)\big], \end{aligned} \tag{19}$$

$$\begin{aligned} \nabla_{\boldsymbol{w}_b} J = 2\big[& (\boldsymbol{x}_a\boldsymbol{x}_b^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_c^{\mathrm{T}} - \boldsymbol{x}_c\boldsymbol{x}_a^{\mathrm{T}})\boldsymbol{w}_a \\ & + (\boldsymbol{x}_c\boldsymbol{x}_c^{\mathrm{T}} + \boldsymbol{x}_a\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_b^{\mathrm{T}})\boldsymbol{w}_b \\ & + (\boldsymbol{x}_c\boldsymbol{x}_b^{\mathrm{T}} - \boldsymbol{x}_a\boldsymbol{x}_c^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_a^{\mathrm{T}})\boldsymbol{w}_c \\ & + (d_a\boldsymbol{x}_c - d_b\boldsymbol{x}_a - d_c\boldsymbol{x}_b)\big], \end{aligned} \tag{20}$$

$$\begin{aligned} \nabla_{\boldsymbol{w}_c} J = 2\big[& (\boldsymbol{x}_a\boldsymbol{x}_c^{\mathrm{T}} - \boldsymbol{x}_b\boldsymbol{x}_a^{\mathrm{T}} - \boldsymbol{x}_c\boldsymbol{x}_b^{\mathrm{T}})\boldsymbol{w}_a \\ & + (\boldsymbol{x}_b\boldsymbol{x}_c^{\mathrm{T}} - \boldsymbol{x}_c\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_a\boldsymbol{x}_b^{\mathrm{T}})\boldsymbol{w}_b \\ & + (\boldsymbol{x}_a\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_c\boldsymbol{x}_c^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_b^{\mathrm{T}})\boldsymbol{w}_c \\ & + (d_a\boldsymbol{x}_b + d_b\boldsymbol{x}_c - d_c\boldsymbol{x}_a)\big]. \end{aligned} \tag{21}$$

Finally, we can obtain the expression for the gradient of $J(n)$ by substituting (19)–(21) into (9),

$$\nabla_{\boldsymbol{w}^*} J(n) = \frac{2}{3} e(n)\boldsymbol{x}^*(n), \tag{22}$$

which yields the update equation in (10), as the term $\frac{2}{3}$ can be absorbed into the step size.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (61331019, 61490690), the China Scholarship Council Postgraduate Scholarship Program (2014), and the UK National Grid.

REFERENCES

[1] I. L. Kantor and A. S. Solodovnikov, *Hypercomplex numbers: An elementary introduction to algebras*. Springer-Verlag, 1989.

[2] J. P. Ward, *Quaternions and Cayley numbers: Algebra and applications*. Dordrecht: Kluwer Academic Publisher, 1997.

[3] N. Le Bihan and S. J. Sangwine, "Quaternion principle component analysis of color images," in *Proceedings of the IEEE International Conference on Image Processing*, Barcelona, Spain, pp. 809–812, Sep. 2003.

[4] S. C. Pei, J. H. Chang, and J. J. Ding, "Commutative reduced biquaternions and their Fourier transform for signal and image processing applications," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 2012–2031, Jul. 2004.

[5] T. A. Ell, "Multi-vector color-image filters," in *Proceedings of the IEEE International Conference on Image Processing*, San Antonio, TX, USA, pp. 245–248, Sep. 2007.

[6] L. Q. Guo, M. Zhu, and X. H. Ge, "Reduced biquaternion canonical transform, convolution and correlation," *Signal Processing*, vol. 91, no. 8, pp. 2147–2153, Aug. 2011.

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

[8] N. Le Bihan, S. Miron, and J. I. Mars, "MUSIC algorithm for vector-sensors array processing using biquaternions," *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4523–4533, Sep. 2007.

[9] J. F. Jiang and J. Q. Zhang, "Geometric algebra of Euclidean 3-space for electromagnetic vector-sensor array processing, part I: Modeling," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 12, pp. 3961–3973, Dec. 2010.

[10] X. F. Gong, Z. W. Liu, and Y. G. Xu, "Coherent source localization: Bicomplex polarimetric smoothing with electromagnetic vector-sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2268–2285, Jul. 2011.

[11] 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, Nov. 2014.

[12] M. B. Hawes and W. Liu, "Design of fixed beamformers based on vector-sensor arrays," *International Journal of Antennas and Propagation*, vol. 2015, 2015.

[13] B. Che Ujang, C. Jahanchahi, C. C. Took, and D. P. Mandic, "Adaptive convex combination approach for the identification of improper quaternion processes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 172–182, Jan. 2014.

[14] F. A. Tobar and D. P. Mandic, "Quaternion reproducing kernel Hilbert spaces: Existence and uniqueness conditions," *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5736–5749, Sep. 2014.

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

[16] J. Navarro-Moreno, R. M. Fernandez-Alcalá, C. C. Took, and D. P. Mandic, "Prediction of wide-sense stationary quaternion random signals," *Signal Processing*, vol. 93, no. 9, pp. 2573–2580, Sep. 2013.

[17] Q. Barthélemy, A. Larue, and J. I. Mars, "About QLMS derivations," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 240–243, Feb. 2014.

[18] R. B. Allenby. *Rings, fields and groups: An introduction to abstract algebra*. Butterworth-Heinemann, 1991.

[19] Async, *OMIC's N-nion's site* [Online]. Available: http://asyncbrain.baf.cz/m/nt/index.htm.

[20] D. Assefa, L. Mansinha, K. F. Tiampo, H. Rasmussen, and K. Abdella, "The trinion Fourier transform of color images," *Signal Processing*, vol. 91, no. 8, pp. 1887–1900, Aug. 2011.

[21] A. van den Bos, "Complex gradient and Hessian," *IEE Proceedings - Vision Image Signal Process*, vol. 141, no. 6, pp. 380–383, Dec. 1994.

[22] M. D. Jiang, W. Liu, and Y. Li, "A general quaternion-valued gradient operator and its applications to computational fluid dynamics and adaptive beamforming," in *Proceedings of the 19th International Conference on Digital Signal Processing*, Hong Kong, China, pp. 821–826, Aug. 2014.

[23] W. Liu, "Antenna array signal processing for a quaternion-valued wireless communication system," in *Proc. the Benjamin Franklin Symposium on Microwave and Antenna Sub-systems (BenMAS)*, Philadelphia, US, September 2014.

[24] M. D. Jiang, Y. Li, and W. Liu, "Properties and applications of a restricted HR gradient operator," *arXiv:1407.5178 [math.OC]*, July 2014.

[25] Google, "RE<C: Surface level wind data collection," *Google Code 2011* [Online]. Available: http://code.google.com/p/google-rec-csp/