This is a repository copy of *Filtering and Tracking with Trinion-Valued Adaptive Algorithms*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/101426/

Version: Accepted Version

Article:

Gou, X.M., Liu, Z.W., Liu, W. orcid.org/0000-0003-2968-2888 et al. (1 more author) (2016) Filtering and Tracking with Trinion-Valued Adaptive Algorithms. Frontiers of Information Technology & Electronic Engineering, 17 (8). pp. 834-840. ISSN 2095-9184

https://doi.org/10.1631/FITEE.1601164

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/

Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Filtering and Tracking with Trinion-Valued Adaptive Algorithms

Xiaoming Gou[1], Zhiwen Liu[1], Wei Liu[†2], Yougen Xu[1]

[1]*School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China.*

[2]*Communications Research Group, Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, UK.*

[†]E-mail: w.liu@sheffield.ac.uk

Abstract: A new model for three-dimensional processes based on the trinion algebra is introduced for the first time. Compared with the pure quaternion model, the trinion model is more compact and computationally more efficient, while having similar or comparable performance in terms of adaptive linear filtering. Moreover, the trinion model can effectively represent the general relationship of state evolution in Kalman filtering, where the pure quaternion model fails. Simulations on real-world wind recordings and synthetic data sets are provided to demonstrate the potentials of this new modeling method.

Key words: Three-dimensional, trinion, least mean squares, Kalman filter.

doi: **Document code:** **CLC number:**

1 Introduction

Multidimensional (m-D) signal processing has a variety of applications and the modeling of multiple variables is carried out traditionally within the real-valued matrix algebra, while in recent years we have observed the successful exploitation of hypercomplex numbers in areas including colour image processing (Pei and Cheng, 1999; Pei *et al.*, 2004; Sangwine and Ell, 2000; Parfieniuk and Petrovsky, 2010; Ell *et al.*, 2014; Liu *et al.*, 2014), vector-sensor array processing (Le Bihan and Mars, 2004; Miron *et al.*, 2006; Le Bihan *et al.*, 2007; Tao, 2013; Tao and Chang, 2014; Zhang *et al.*, 2014; Hawes and Liu, 2015; Jiang *et al.*, 2016a,b), and quaternion-valued wireless communications (Zetterberg and Brandstrom, 1977; Isaeva and Sarytchev, 1995; Liu, 2014). The most widely used hypercomplex numbers are quaternions, with rigorous physical interpretation for 3-D and 4-D rotational problems (Kantor *et al.*, 1989; Ward, 1997). In particular, for the 3-D case, such as 3-D altitude and 3-D wind speed, they are usually modeled with pure quaternions in literature (Jiang *et al.*, 2014; Jahanchahi and Mandic, 2014; Talebi and Mandic, 2015).

However, pure quaternions do not belong to a mathematical ring (Allenby, 1991), as the product of two pure quaternions is no longer a pure quaternion in general. This could indicate redundant computations. For instance, the adaptive algorithms for 3-D signal filtering, which are initialised with pure quaternions (Jiang *et al.*, 2014; Quentin *et al.*, 2014), have to update themselves with full quaternions and truncate their results from a full quaternion to a pure quaternion. In terms of the hypercomplex multiplication alone, 16 real-valued multiplications and 12 real-valued additions are required to calculate the product of two full quaternions, while these two quantities will be reduced to 9 and 6, respectively, for two numbers of a 3-D ring. Furthermore, we will see in this paper that pure quaternions can not be used to model the general 3-D tracking problems.

As a solution, in this paper we introduce a new type of hypercomplex number termed trinion for 3-D adaptive filtering and tracking. Trinions form a 3-D ring and are commutative by definition (Assefa *et al.*, 2011), which implies that the trinion algebra could be a competitive candidate for modeling 3-D processes. In our first contribution, a class of trinion-valued least mean squares (LMS) algorithms is developed to show that trinions are computationally more efficient than quaternion algebra for 3-D adaptive filtering applications. Secondly, we extend the classic Kalman filter (Chui and Chen, 1991; Li *et al.*, 2015) into the trinion domain for efficient and

©Zhejiang University and Springer-Verlag Berlin Heidelberg 2015

effective 3-D tracking. We will see that for the most general case, a pure quaternion model will not work, while trinion algebra provides a convenient and compact solution. For the first contribution, the augmented second-order statistics are also considered (Adalı and Schreier, 2014).

This paper is organised as follows. A brief introduction to trinions and the augmented trinion statistics is provided in Section II. The trinion-valued LMS algorithm and Kalman filter are derived in Section III. Simulation results are provided in Section IV, followed by conclusions in Section V.

2 Trinions

A trinion v is a hypercomplex number comprising one real part and two imaginary parts,

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

with the two imaginary units $\imath$ and $\jmath$ satisfying (Assefa *et al.*, 2011)

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

from which it can be observed that trinions are commutative.

The following is a brief list of properties of trinions involved in formulating algorithms.

1. The (Euclidean) modulus of v is expressed as

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

and we define the conjugate of v as

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

so that $|v|^2 = \Re(vv^*)$, where $\Re(\cdot)$ denotes the real part. As a result, for two trinions v_1 and v_2, we have $(v_1v_2)^* = v_1^*v_2^*$.

2. The complete information of second-order statistics of a trinion-valued multivariate variable (in a vector form) $\mathbf{v} = \mathbf{v}_a + \imath\mathbf{v}_b + \jmath\mathbf{v}_c$ is contained in the following six real-valued covariance matrices:

$$\begin{aligned}&\mathbf{C}_{\mathbf{v}_\theta\mathbf{v}_\phi} = E\{\mathbf{v}_\theta\mathbf{v}_\phi^{\mathrm{T}}\} \ ,\\ &(\theta,\phi) \in \{(a,a),(b,b),(c,c),(a,b),(b,c),(c,a)\} \ .\end{aligned} \tag{5}$$

Equivalently, these matrices can be represented by three trinion-valued covariance matrices,

$$\begin{aligned}\mathbf{C}_{\mathbf{v}\mathbf{v}} &= E\{\mathbf{v}\mathbf{v}^{\mathrm{H}}\},\\ \mathbf{C}_{\mathbf{v}\mathbf{v}^\imath} &= E\{\mathbf{v}\mathbf{v}^{\imath\mathrm{H}}\},\\ \mathbf{C}_{\mathbf{v}\mathbf{v}^\jmath} &= E\{\mathbf{v}\mathbf{v}^{\jmath\mathrm{H}}\},\end{aligned} \tag{6}$$

where $(\cdot)^{\mathrm{H}}$ denotes Hermitian transpose and we have defined two additional mappings (for short-hand notions only) of $\mathbf{v}$ as

$$\mathbf{v}^\imath = \mathbf{v}_b - \imath\mathbf{v}_a - \jmath\mathbf{v}_c \ , \mathbf{v}^\jmath = \mathbf{v}_c - \imath\mathbf{v}_b - \jmath\mathbf{v}_a \ . \tag{7}$$

The real-valued covariance matrices can be easily retrieved from the trinion-valued ones, namely,

$$\begin{aligned}\mathbf{C}_{\mathbf{v}_a\mathbf{v}_a} &= \frac{1}{2}\Re(\mathbf{C}_{\mathbf{v}\mathbf{v}} + \jmath\mathbf{C}_{\mathbf{v}\mathbf{v}^\imath}) \ ,\\ \mathbf{C}_{\mathbf{v}_b\mathbf{v}_b} &= \frac{1}{2}\Re(\imath\mathbf{C}_{\mathbf{v}\mathbf{v}^\jmath} - \jmath\mathbf{C}_{\mathbf{v}\mathbf{v}^\imath}) \ ,\\ \mathbf{C}_{\mathbf{v}_c\mathbf{v}_c} &= \frac{1}{2}\Re(\mathbf{C}_{\mathbf{v}\mathbf{v}} - \imath\mathbf{C}_{\mathbf{v}\mathbf{v}^\jmath}) \ ,\\ \mathbf{C}_{\mathbf{v}_a\mathbf{v}_b} &= \frac{1}{2}\Re(\mathbf{C}_{\mathbf{v}\mathbf{v}^\imath} + \jmath\mathbf{C}_{\mathbf{v}\mathbf{v}^\jmath}) \ ,\\ \mathbf{C}_{\mathbf{v}_b\mathbf{v}_c} &= \frac{1}{2}\Re(\imath\mathbf{C}_{\mathbf{v}\mathbf{v}} - \jmath\mathbf{C}_{\mathbf{v}\mathbf{v}^\jmath}) \ ,\\ \mathbf{C}_{\mathbf{v}_c\mathbf{v}_a} &= \frac{1}{2}\Re(\mathbf{C}_{\mathbf{v}\mathbf{v}^\imath} - \imath\mathbf{C}_{\mathbf{v}\mathbf{v}}) \ .\end{aligned} \tag{8}$$

3. The calculation of trinion-valued gradient is important for adaptive algorithm derivation. In the complex domain, the gradient is based on the assumption that a function of variable z is a function of z and its conjugate (Brandwood, 1983; Adalı and Schreier, 2014). A similar prerequisite in the quaternion domain is that a function of variable q is a function of q and its three involutions (Jiang *et al.*, 2014). The same concept would fail in the trinion domain, since the trinion involution does not exist in general, at least to our best knowledge. Hence, we simply follow the form of the complex-valued gradient and define the trinion-valued gradients of a function $f(\mathbf{v})$ with respect to the variable $\mathbf{v}$ and its conjugate by

$$\begin{aligned}\nabla_{\mathbf{v}}f &= \frac{1}{3}\big(\nabla_{\mathbf{v}_a}f - \jmath\nabla_{\mathbf{v}_b}f - \imath\nabla_{\mathbf{v}_c}f\big) \ ,\\ \nabla_{\mathbf{v}^*}f &= \frac{1}{3}\big(\nabla_{\mathbf{v}_a}f + \imath\nabla_{\mathbf{v}_b}f + \jmath\nabla_{\mathbf{v}_c}f\big) \ .\end{aligned} \tag{9}$$

Since trinions are commutative, the imaginary units $\imath$ and $\jmath$ can be on any side of the real-valued gradients.

The derivatives of some simple functions can be calculated, for example,

$$\frac{\partial v}{\partial v} = \frac{\partial v^*}{\partial v^*} = 1 \ , \ \frac{\partial v}{\partial v^*} = \frac{\partial v^*}{\partial v} = \frac{1-\imath+\jmath}{3} \ , \quad (10)$$

$$\frac{\partial \Re[Tr(\mathbf{VW})]}{\partial \mathbf{V}} = \frac{1}{3}\mathbf{W}^{\mathrm{T}} \ , \ \frac{\partial \Re[Tr(\mathbf{WV}^{\mathrm{H}})]}{\partial \mathbf{V}} = \frac{1}{3}\mathbf{W}^* \ ,$$
$$\frac{\partial \Re[Tr(\mathbf{VWV}^{\mathrm{H}})]}{\partial \mathbf{V}} = \frac{1}{3}\mathbf{V}^*(\mathbf{W}^* + \mathbf{W}^{\mathrm{T}}) \ . \quad (11)$$

3 Trinion-Valued Filtering Algorithms

3.1 Trinion-Valued LMS Adaptive Algorithm

We consider the filtering of a tri-variate signal based on the LMS principle (Haykin and Widrow, 2003). The error is expressed as

$$e(n) = d(n) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n) \ , \quad (12)$$

where $d(n)$ is the reference signal, $\mathbf{w}(n)$ is the weight vector, $\mathbf{x}(n) = [x(n), x(n-1), \cdots, x(n-L+1)]^{\mathrm{T}}$ is the filter input, and L is the filter length. The cost function is given by

$$J(n) = |e(n)|^2 \ . \quad (13)$$

According to the steepest descent method, we need to calculate the following gradient (details can be found in Appendix A)

$$\begin{aligned}\nabla_{\mathbf{w}^*} J(n) &= \frac{1}{3}\left[\nabla_{\mathbf{w}_a} J(n) + \imath \nabla_{\mathbf{w}_b} J(n) + \jmath \nabla_{\mathbf{w}_c} J(n)\right] \\ &= \frac{2}{3} e(n)\mathbf{x}^*(n) \ ,\end{aligned} \quad (14)$$

yielding the following update equation for the weight vector

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}^*(n) \ , \quad (15)$$

where μ is the step size with the scale factor $\frac{2}{3}$ absorbed into it. This LMS-like algorithm is termed as the Trinion-valued LMS (TLMS) algorithm.

To account for the complete second-order statistics, the augmented filtering structure is required, which gives an output $y(n)$ as

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

where $\mathbf{x}^{\mathrm{aug}}(n) = [\mathbf{x}(n); \mathbf{x}^{\imath}(n); \mathbf{x}^{\jmath}(n)]$ and $\mathbf{w}^{\mathrm{aug}}(n) = [\mathbf{w}_1; \mathbf{w}_2; \mathbf{w}_3]$. Similarly, we have the following update equation for the augmented weight vector

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

where ρ is the step size. We call this algorithm the Augmented Trinion-valued LMS (ATLMS) algorithm.

The computational complexities for each update of the weight vector of the LMS-like filtering algorithms in the trinion and quaternion domains are shown in Table I, where the quaternion-valued LMS (QLMS) algorithm and the augmented QLMS algorithm are based on the result in (Jiang *et al.*, 2014; Quentin *et al.*, 2014; Tao and Chang, 2014). Clearly, the trinion model has a much lower complexity than the quaternion model.

Table 1 Computions needed per update of the weight vector

Algorithm	Real Multiplications	Real Additions
TLMS	$9L+3$	$9L$
Augmented TLMS	$27L+3$	$27L$
QLMS	$16L+4$	$16L$
Augmented QLMS	$64L+4$	$64L$

3.2 Trinion-valued Kalman Filter

In this subsection we focus on the Kalman estimate of a tri-variate vector state $\mathbf{x}_k$ which evolves by the following trinion-valued model:

$$\mathbf{x}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \boldsymbol{\omega}_k \quad (18)$$

where $\mathbf{A}_k$ is the state transition matrix, $\mathbf{u}_k$ is the input controlled by $\mathbf{B}_k$, and $\boldsymbol{\omega}_k$ is the state noise. Note that if the process is modelled with pure quaternions, the state transition matrix $\mathbf{A}_k$ must be real-valued so that all states evolved are pure quaternion-valued and all three real-valued sub-states evolve independently with each other, which would be unrealistic in practice. In comparison, the trinion-valued state model is not subject to this constraint and hence more flexible in modelling tri-variate states.

The observation $\mathbf{z}_k$ of the state $\mathbf{x}_k$ is given by

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \boldsymbol{\upsilon}_k \ , \quad (19)$$

where $\mathbf{H}$ is the observation matrix and $\boldsymbol{\upsilon}_k$ is the measurement noise. Both $\boldsymbol{\omega}_k$ and $\boldsymbol{\upsilon}_k$ are assumed to be zero-mean white-Gaussian, i.e. $\boldsymbol{\omega}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ and

$\boldsymbol{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. The *a priori* and *a posteriori* state estimates are expressed as

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \ , \tag{20}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_{k|k-1} \right) \ , \tag{21}$$

respectively, where $\hat{\mathbf{x}}_{k-1|k-1}$ is the previous state estimate, $\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}$ represents the innovation, and $\mathbf{K}_k$ is the unknown Kalman gain matrix and can be found by minimizing the power of the error

$$\begin{aligned} \mathbf{e}_{k|k} \quad = \quad & \mathbf{x}_k - \hat{\mathbf{x}}_{k|k} = \mathbf{x}_k - \\ & \left[\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \right) \right] , \end{aligned} \tag{22}$$

which is

$$\begin{aligned} & E\left\{ \|\mathbf{e}_{k|k}\|^2 \right\} \\ =& \Re \left\{ Tr \left[\mathrm{cov} \left(\mathbf{x}_k - \left(\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \right) \right) \right) \right] \right\} \\ =& \Re \left\{ Tr \left[\mathrm{cov} \left[\left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right) \left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} - \mathbf{K}_k \mathbf{v}_k \right) \right] \right] \right\} \ . \end{aligned} \tag{23}$$

Since the noise is independent of the states, we have

$$\begin{aligned} & E\left\{ \|\mathbf{e}_{k|k}\|^2 \right\} \\ =& \Re \Big\{ Tr \Big[\left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right) \mathrm{cov} \left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \right) \\ & \cdot \left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right)^{\mathrm{H}} + \mathbf{K}_k \mathrm{cov} \left(\boldsymbol{v}_k \right) \mathbf{K}_k^{\mathrm{H}} \Big] \Big\} \ , \end{aligned} \tag{24}$$

where the matrix $\mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$ is known as the *a priori* error covariance matrix $\mathbf{P}_k$, and it follows

$$\begin{aligned} E\left\{ \|\mathbf{e}_{k|k}\|^2 \right\} = \Re \Big\{ \Big[& \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H} \mathbf{P}_{k|k-1} \\ & - \mathbf{P}_{k|k-1} \mathbf{H}^{\mathrm{H}} \mathbf{K}_k^{\mathrm{H}} + \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{\mathrm{H}} \Big] \Big\} \ , \end{aligned} \tag{25}$$

where $\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^{\mathrm{H}} + \mathbf{R}_k$. Taking the partial derivative of $E\{||\mathbf{e}_{k|k}||^2\}$ with respect to $\mathbf{K}_k$ and setting it to zero, we have

$$\begin{aligned} \frac{\partial E\left\{ \|\mathbf{e}_{k|k}\|^2 \right\}}{\partial \mathbf{K}_k} =& -\frac{\partial \Re \left\{ Tr \left[\mathbf{K}_k \mathbf{H} \mathbf{P}_{k|k-1} \right] \right\}}{\partial \mathbf{K}_k} \\ & - \frac{\partial \Re \left\{ Tr \left[\mathbf{P}_{k|k-1} \mathbf{H}^{\mathrm{H}} \mathbf{K}_k^{\mathrm{H}} \right] \right\}}{\partial \mathbf{K}_k} \\ & + \frac{\partial \Re \left\{ Tr \left[\mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{\mathrm{H}} \right] \right\}}{\partial \mathbf{K}_k} = \mathbf{0} \ , \end{aligned} \tag{26}$$

which yields (details of the derivation are provided in Appendix B)

$$\mathbf{K}_k = \frac{1}{2} \mathbf{P}_{k|k-1} \left(\mathbf{H}^{\mathrm{H}} + \mathbf{H}^{\mathrm{T}} \right) \mathbf{S}_k^{-1} \ . \tag{27}$$

Since it is assumed that the noise is independent of the states, we have

$$\begin{aligned} \mathbf{P}_{k|k-1} &= \mathrm{cov} \left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \right) \\ &= \mathbf{A} \mathbf{P}_{k-1|k-1} \mathbf{A}^{\mathrm{H}} + \mathbf{Q}_k, \end{aligned} \tag{28}$$

and subsequently we obtain the updated covariance matrix as

$$\begin{aligned} \mathbf{P}_k =& \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H} \mathbf{P}_{k|k-1} \\ & - \mathbf{P}_{k|k-1} \mathbf{H}^{\mathrm{H}} \mathbf{K}_k^{\mathrm{H}} + \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{H} \end{aligned} \tag{29}$$

This Kalman-like filter is termed as the Trinion-valued Kalman Filter (TKF) and is summerised in Table II.

Table 2 Trinion-valued Kalman filter

Predict
$\hat{\mathbf{x}}_{k\|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1\|k-1} + \mathbf{B}_k \mathbf{u}_k$ $\mathbf{P}_{k\|k-1} = \mathbf{A} \mathbf{P}_{k-1\|k-1} \mathbf{A}^{\mathrm{H}} + \mathbf{Q}_k$
Update
$\mathbf{S}_k = \mathbf{H} \mathbf{P}_{k\|k-1} \mathbf{H}^{\mathrm{H}} + \mathbf{R}_k$ $\mathbf{K}_k = \frac{1}{2} \mathbf{P}_{k\|k-1} (\mathbf{H}^{\mathrm{H}} + \mathbf{H}^{\mathrm{T}}) \mathbf{S}_k^{-1}$ $\hat{\mathbf{x}}_{k\|k-1} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_{k\|k-1})$ $\mathbf{P}_k = \mathbf{P}_{k\|k-1} - \mathbf{K}_k \mathbf{H} \mathbf{P}_{k\|k-1}$ $-\mathbf{P}_{k\|k-1} \mathbf{H}^{\mathrm{H}} \mathbf{K}_k^{\mathrm{H}} + \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{H}$

4 Simulated Results

In this section, simulation results are provided to demonstrate the performance of the derived algorithms.

First, simulations are performed using the TLMS and ATLMS algorithms for wind speed prediction based on data from the surface-level anemometer readings provided by Google (Google), and the wind speed measured on May 31, 2011 is used as an example.

The learning curves averaged over 150 trials of the proposed algorithms are shown in Fig. 1, compared with the quaternion-based QLMS and AQLMS algorithms, where the step size is 6×10^{-5}, the filter length is 8, the prediction step is 1, and all algorithms are initialised with an all-zero filter coefficients. It can be observed that both augmented algorithms (AQLMS and ATLMS) have a similar faster convergence rate than the original ones (QLMS and TLMS), since they have taken the complete second-order statistics into consideration. Besides, the proposed TLMS algorithm has a slightly better performance than the QLMS algorithm, while the ATLMS algorithm is comparable with the AQLMS algorithm. However, we should bear in mind that the proposed trinion-based algorithms have a much lower computational complexity, as shown in Table I.

In the next, we test the TKF algorithm with synthetic data generated by the following model:

$$
\begin{aligned}
\mathbf{x}_k &= \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} \\
&= \begin{bmatrix} 1+0.3\imath+0.3\jmath & 0.1+0.2\imath+0.1\jmath \\ -0.1 & 1+0.1\imath+0.2\jmath \end{bmatrix} \mathbf{x}_{k-1} + \boldsymbol{\omega}_k, \\
\mathbf{z}_k &= \begin{bmatrix} 1+0.7\imath+0.5\jmath & 0.5+0.4\imath+0.1\jmath \\ 0.2+0.3\imath+0.4\jmath & 1+0.2\imath+0.5\jmath \end{bmatrix} \mathbf{x}_k + \boldsymbol{\upsilon}_k, \\
\boldsymbol{\omega}_k &\sim \mathcal{N}(\mathbf{0},\mathbf{Q}) \ , \ \boldsymbol{\upsilon}_k \ \sim \mathcal{N}(\mathbf{0},\mathbf{R}), \ \mathbf{Q}=\mathbf{R}=\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}, \\
\mathbf{x}_0 &= \begin{bmatrix} 2.5+2\imath+\jmath \\ 3\imath+4\jmath \end{bmatrix},
\end{aligned}
\tag{30}
$$

where we can see that the three sub-state vectors $\mathbf{x}_{ka}, \mathbf{x}_{kb}, \mathbf{x}_{kc}$ evolve dependently, and the observation $\mathbf{z}_k$ is a linear mixture of them. The filtered results are plotted in Fig. 2 (for $\mathbf{x}_{ka}$), Fig. 3 (for $\mathbf{x}_{kb}$), and Fig. 4 (for $\mathbf{x}_{kc}$). The errors in modulus before and after filtering are depicted in Fig. 5. We can observe from the results that TKF can track the system state $\mathbf{x}_k$ effectively.

5 Conclusion

A trinion-valued model for filtering and tracking of three-dimensional signals has been proposed, with corresponding algorithms derived, including two LMS-type algorithms (trinion-valued LMS and its augmented version) for adaptive filtering, and a Kalman filtering algorithm for tracking. Simulation results have shown that the trinion model is a competitive candidate for

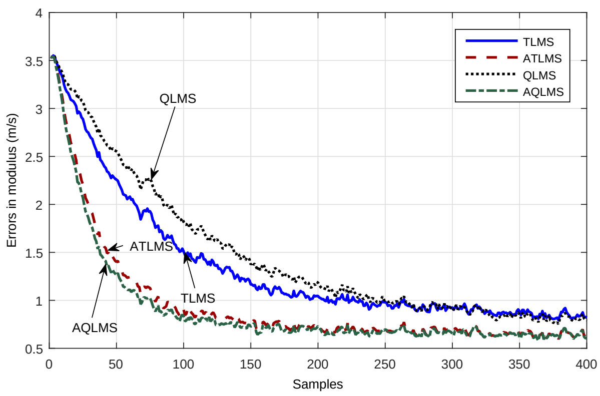

Figure 1 Averaged learning curves.

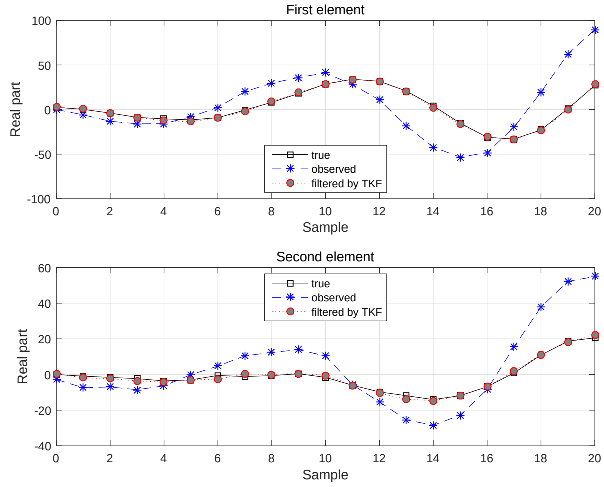

Figure 2 Filtered results from TKF: real part.

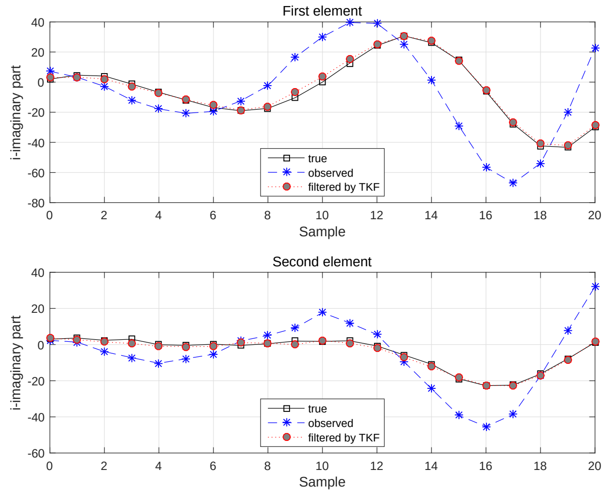

Figure 3 Filtered results from TKF: $\imath$-imaginary part.

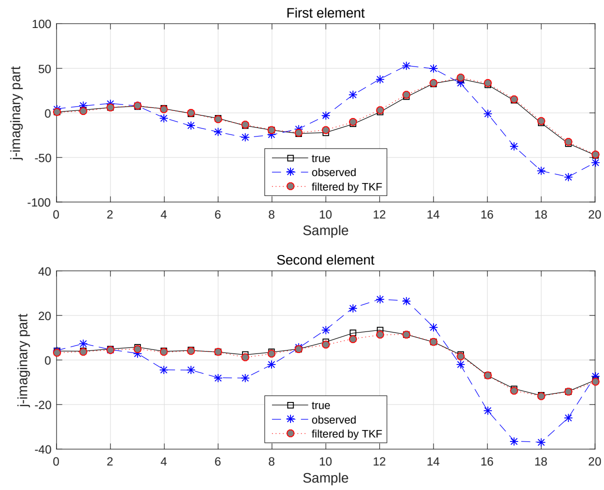

Figure 4 Filtered results from TKF: $\jmath$-imaginary part.

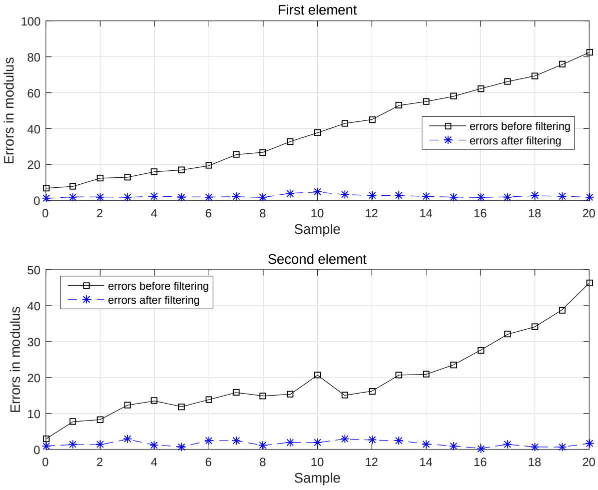

Figure 5 Errors in modulus before and after filtering.

duced computational complexity (related to its compactness) and effective modeling of more complicated three-dimensional processes (related to its closure property).

A Calculation of the gradient in (14)

We can expand the cost function $J(n)$ as

$$\begin{aligned} J = {} & (d_a - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_a + \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_c + \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_b)^2 \\ & + (d_b - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_b - \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_a + \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_c)^2 \\ & + (d_c - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_c - \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_b - \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_a)^2, \end{aligned} \tag{31}$$

where we have dropped the time index for convenience. Then we can calculate the gradients with respect to each part of the weight vector, i.e.

$$\begin{aligned} \nabla_{\boldsymbol{w}_a} J = 2\big[& (\boldsymbol{x}_a\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_b^{\mathrm{T}} + \boldsymbol{x}_c\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_a \\ & +(\boldsymbol{x}_b\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_c\boldsymbol{x}_b^{\mathrm{T}} - \boldsymbol{x}_a\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_b \\ & +(\boldsymbol{x}_c\boldsymbol{x}_a^{\mathrm{T}} - \boldsymbol{x}_a\boldsymbol{x}_b^{\mathrm{T}} - \boldsymbol{x}_b\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_c \\ & -(d_a\boldsymbol{x}_a + d_b\boldsymbol{x}_b + d_c\boldsymbol{x}_c)\big] \end{aligned} \tag{32}$$

$$\begin{aligned} \nabla_{\mathbf{w}_b} J = 2\big[& (\mathbf{x}_a\mathbf{x}_b^{\mathrm{T}} + \mathbf{x}_b\mathbf{x}_c^{\mathrm{T}} - \mathbf{x}_c\mathbf{x}_a^{\mathrm{T}})\mathbf{w}_a \\ & +(\mathbf{x}_c\mathbf{x}_c^{\mathrm{T}} + \mathbf{x}_a\mathbf{x}_a^{\mathrm{T}} + \mathbf{x}_b\mathbf{x}_b^{\mathrm{T}})\mathbf{w}_b \\ & +(\mathbf{x}_c\mathbf{x}_b^{\mathrm{T}} - \mathbf{x}_a\mathbf{x}_c^{\mathrm{T}} + \mathbf{x}_b\mathbf{x}_a^{\mathrm{T}})\mathbf{w}_c \\ & +(d_a\mathbf{x}_c - d_b\mathbf{x}_a - d_c\mathbf{x}_b)\big] \end{aligned} \tag{33}$$

$$\begin{aligned} \nabla_{\mathbf{w}_c} J = 2\big[& (\mathbf{x}_a\mathbf{x}_c^{\mathrm{T}} - \mathbf{x}_b\mathbf{x}_a^{\mathrm{T}} - \mathbf{x}_c\mathbf{x}_b^{\mathrm{T}})\mathbf{w}_a \\ & +(\mathbf{x}_b\mathbf{x}_c^{\mathrm{T}} - \mathbf{x}_c\mathbf{x}_a^{\mathrm{T}} + \mathbf{x}_a\mathbf{x}_b^{\mathrm{T}})\mathbf{w}_b \\ & +(\mathbf{x}_a\mathbf{x}_a^{\mathrm{T}} + \mathbf{x}_c\mathbf{x}_c^{\mathrm{T}} + \mathbf{x}_b\mathbf{x}_b^{\mathrm{T}})\mathbf{w}_c \\ & +(d_a\mathbf{x}_b + d_b\mathbf{x}_c - d_c\mathbf{x}_a)\big] \end{aligned} \tag{34}$$

Finally, the gradient of $J(n)$ is obtained by merging (32)–(34) into (14),

$$\nabla_{\mathbf{w}^*} J(n) = \frac{2}{3} e(n)\mathbf{x}^*(n) \ . \tag{35}$$

B Calculation of the Kalman gain matrix

We know from (11) and (26) that

$$-\frac{1}{3}\mathbf{P}_{k|k-1}^{\mathrm{T}}\mathbf{H}^{\mathrm{H}} - \frac{1}{3}\mathbf{P}_{k|k-1}^{*}\mathbf{H}^{\mathrm{T}} + \frac{1}{3}\mathbf{K}_k\left(\mathbf{S}_k^* + \mathbf{S}_k^{\mathrm{T}}\right) = \mathbf{0} \ . \tag{36}$$

Since $\mathbf{S}_k$ and $\mathbf{P}_{k|k-1}$ are both Hermitian, i.e.

$$\mathbf{S}_k^* = \mathbf{S}_k^{\mathrm{T}} \ , \ \mathbf{P}_{k|k-1}^* = \mathbf{P}_{k|k-1}^{\mathrm{T}} \ , \tag{37}$$

we have

$$-\frac{1}{3}\mathbf{P}_{k|k-1}^{*}\mathbf{H}^{\mathrm{H}} - \frac{1}{3}\mathbf{P}_{k|k-1}^{*}\mathbf{H}^{\mathrm{T}} + \frac{2}{3}\mathbf{K}_k\mathbf{S}_k^* = \mathbf{0} \ , \tag{38}$$

which yields (27).

References

Adalı, T., Schreier, P. J., 2014. optimization and estimation of complex-valued signals: theory and applications in filtering and blind source separation. *IEEE Signal Processing Magazine*, **31**(5):112-128.

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

Assefa, D., Mansinha, L., Tiampo, K. F., Rasmussen, H., Abdella, K., 2011. the trinion Fourier transform of color images. *Signal Processing*, **91**(8):1897–1900.

Brandwood, D.H., 1983. a complex gradient operator and its application in adaptive array theory. *IEE Proceedings H (Microwaves, Optics and Antennas)*, **130**(1):11–16.

Chui, C. K., Chen, G. R., 1991. Kalman filtering. Springer Verlag, Berlin.

Ell, T. A., Le Bihan, N., Sangwine, S. J., 2014. Quaternion fourier transforms for signal and image processing. Wiley.

Surface level wind data collection, http://code.google.com/p/google-rec-csp/downloads/list.

Hawes, M. B., Liu, W., 2015. design of fixed beamformers based on vector-sensor arrays. *International Journal of Antennas and Propagation*, **2015**.

Haykin, S., Widrow, B., 2003. Least-mean-square adaptive filters. John Wiley & Sons, New York.

Isaeva, O. M., Sarytchev, V. A., 1995. Quaternion presentations polarization state. Proc. 2nd IEEE Topical Symposium of Combined Optical-Microwave Earth and Atmosphere Sensing, Atlanta, US, p.195–196.

Jahanchahi, C., Mandic, D. P., 2014. a class of quaternion kalman filters. *IEEE Transactions on Neural Networks and Learning Systems*, **25**(3):533–544.

Jiang, M. D., Li, Y., Liu, W., 2016a. properties of a general quaternion-valued gradient operator and its application to signal processing. *Frontiers of Information Technology & Electronic Engineering*, **17**:83–95.

Jiang, M. D., Liu, W., Li, Y., 2014. A general quaternion-valued gradient operator and its applications to computational fluid dynamics and adaptive beamforming. Proc. of the International Conference on Digital Signal Processing, Hong Kong.

Jiang, M. D., Liu, W., Li, Y., 2016b. adaptive beamforming for vector-sensor arrays based on reweighted zero-attracting quaternion-valued LMS algorithm. *IEEE Trans. on Circuits and Systems II: Express Briefs*, **63**:274–278.

Kantor, I., Solodovnikov, A. S., Shenitzer, A., 1989. Hypercomplex numbers: an elementary introduction to algebras. Springer Verlag, New York.

Le Bihan, N., Mars, J., 2004. singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing. *Signal Processing*, **84**(7):1177–1199.

Le Bihan, N., Miron, S., Mars, J. I., 2007. MUSIC algorithm for vector-sensors array using biquaternions. *IEEE Transactions on Signal Processing*, **55**(9):4523–4533.

Li, T. C., Villarrubia, G., Sun, S. D., Corchado, J. M., Bajo, J., 2015. resampling methods for particle filtering: identical distribution, a new method, and comparable study. *Frontiers of Information Technology & Electronic Engineering*, **16**:969–984.

Liu, H., Zhou, Y. L., Gu, Z. P., 2014. inertial measurement unit-camera calibration based on incomplete inertial sensor information. *Journal of Zhejiang University SCIENCE C*, **15**:999–1008.

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

Miron, S., Le Bihan, N., Mars, J. I., 2006. quaternion-MUSIC for vector-sensor array processing. *IEEE Transactions on Signal Processing*, **54**(4):1218–1229.

Parfieniuk, M., Petrovsky, A., 2010. inherently lossless structures for eight- and six-channel linear-phase paraunitary filter banks based on quaternion multipliers. *Signal Processing*, **90**(6):1755–1767.

Pei, S. C., Chang, J. H., Ding, J. J., 2004. commutative reduced biquaternions and their fourier transform for signal and image processing applications. *IEEE Transactions on Signal Processing*, **52**(7):2012–2031.

Pei, S.C., Cheng, C.M., 1999. color image processing by using binary quaternion-moment-preserving thresholding technique. *IEEE Transactions on Image Processing*, **8**(5):614-628.

Quentin, B., Larue, A., Mars, J., 2014. about QLMS derivations. *IEEE Signal Processing Letters*, **21**(2):240–243.

Sangwine, S. J., Ell, T. A., 2000. The discrete fourier transform of a colour image. Proc. Image Processing II: Mathematical Methods, Algorithms and Applications, p.430–441.

Talebi, S. P., Mandic, D. P., 2015. A quaternion frequency estimator for three-phase power systems. Proc. the IEEE International Conference on Acoustics, Speech and Signal Processing, Brisbane, Australia, p.3956–3960.

Tao, J.-W., 2013. performance analysis for interference and noise canceller based on hypercomplex and spatio-temporal-polarisation processes. *IET Radar, Sonar Navigation*, **7**(3):277–286.

Tao, J. W., Chang, W. X., 2014. adaptive beamforming based on complex quaternion processes. *Mathematical Problems in Engineering*, **2014**.

Ward, J. P., 1997. Quaternions and cayley numbers: Algebra and applications. Kluwer, Dordrecht.

Zetterberg, L. H., Brandstrom, H., 1977. codes for combined phase and amplitude modulated signals in a four-dimensional space. *IEEE Transactions on Communications*, **25**(29):943–950.

Zhang, X. R., Liu, W., Xu, Y. G., Liu, Z. W., 2014. quaternion-valued robust adaptive beamformer for electromagnetic vector-sensor arrays with worst-case constraint. *Signal Processing*, **104**:274–283.