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*#

Xiao-ming GOU[1], Zhi-wen LIU[1], Wei LIU[†‡2], You-gen 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

Received Apr. 13, 2016; Revision accepted June 24, 2016; Crosschecked July 11, 2016

Abstract: A new model for three-dimensional processes based on the trinion algebra is introduced for the first time. Compared to 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 potential of this new modeling method.

Key words: Three-dimensional processes, Trinion, Least mean squares, Kalman filter
http://dx.doi.org/10.1631/FITEE.1601164 **CLC number:** TN911.7

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 color image processing (Pei and Cheng, 1999; Sangwine *et al.*, 2000; Pei *et al.*, 2004; 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; 2016b), 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 and Solodovnikov, 1989; Ward, 1997). In particular, for 3-D cases, such as 3-D altitude and 3-D wind speed, they are usually modeled with pure quaternions in the literature (Jahanchahi and Mandic, 2014; Jiang *et al.*, 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 initialized with pure quaternions (Barthélemy *et al.*, 2014; Jiang *et al.*,

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61331019 and 61490691), the China Scholarship Council Postgraduate Scholarship Program (2014), and the National Grid (UK)

A preliminary version was presented at the IEEE International Conference on Digital Signal Processing, Singapore, July 21–24, 2015

ORCID: Wei LIU, http://orcid.org/0000-0003-2968-2888

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 study that pure quaternions cannot be used to model the general 3-D tracking problems.

As a solution, in this study we introduce a new type of hypercomplex number, termed 'trinion', for 3-D adaptive filtering and tracking (a brief introduction has been presented in Gou *et al.* (2015)). 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. Second, we extend the classic Kalman filter (Chui and Chen, 1991; Li *et al.*, 2015) into the trinion domain for efficient and 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 (Adali and Schreier, 2014).

2 Trinions

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

$$v = v_a + \mathrm{i}v_b + \mathrm{j}v_c, \tag{1}$$

with the two imaginary units i and j satisfying (Assefa *et al.*, 2011)

$$\mathrm{i}^2 = \mathrm{j},\ \mathrm{ij} = \mathrm{ji} = -1,\ \mathrm{j}^2 = -\mathrm{i}, \tag{2}$$

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

$$v^* = v_a - \mathrm{j}v_b - \mathrm{i}v_c, \tag{4}$$

so that $|v|^2 = \mathrm{R}(vv^*)$, where $\mathrm{R}(\cdot)$ 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 the second-order statistics of a trinion-valued multivariate variable (in a vector form) $\boldsymbol{v} = \boldsymbol{v}_a + \mathrm{i}\boldsymbol{v}_b + \mathrm{j}\boldsymbol{v}_c$ is contained in the following six real-valued covariance matrices:

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

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

$$\begin{cases}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}} = E\{\boldsymbol{v}\boldsymbol{v}^{\mathrm{H}}\},\\ \boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{i}}} = E\{\boldsymbol{v}\boldsymbol{v}^{\mathrm{iH}}\},\\ \boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{j}}} = E\{\boldsymbol{v}\boldsymbol{v}^{\mathrm{jH}}\},\end{cases} \tag{6}$$

where $(\cdot)^{\mathrm{H}}$ denotes the Hermitian transpose and we have defined two additional mappings (for shorthand notions only) of $\boldsymbol{v}$ as

$$\begin{cases}\boldsymbol{v}^{\mathrm{i}} = \boldsymbol{v}_b - \mathrm{i}\boldsymbol{v}_a - \mathrm{j}\boldsymbol{v}_c,\\ \boldsymbol{v}^{\mathrm{j}} = \boldsymbol{v}_c - \mathrm{i}\boldsymbol{v}_b - \mathrm{j}\boldsymbol{v}_a.\end{cases} \tag{7}$$

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

$$\begin{cases}\boldsymbol{C}_{\boldsymbol{v}_a\boldsymbol{v}_a} = \frac{1}{2}\mathrm{R}(\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}} + \mathrm{j}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{i}}}),\\ \boldsymbol{C}_{\boldsymbol{v}_b\boldsymbol{v}_b} = \frac{1}{2}\mathrm{R}(\mathrm{i}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{j}}} - \mathrm{j}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{i}}}),\\ \boldsymbol{C}_{\boldsymbol{v}_c\boldsymbol{v}_c} = \frac{1}{2}\mathrm{R}(\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}} - \mathrm{i}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{j}}}),\\ \boldsymbol{C}_{\boldsymbol{v}_a\boldsymbol{v}_b} = \frac{1}{2}\mathrm{R}(\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{i}}} + \mathrm{j}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{j}}}),\\ \boldsymbol{C}_{\boldsymbol{v}_b\boldsymbol{v}_c} = \frac{1}{2}\mathrm{R}(\mathrm{i}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}} - \mathrm{j}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{j}}}),\\ \boldsymbol{C}_{\boldsymbol{v}_c\boldsymbol{v}_a} = \frac{1}{2}\mathrm{R}(\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}^{\mathrm{i}}} - \mathrm{i}\boldsymbol{C}_{\boldsymbol{v}\boldsymbol{v}}).\end{cases} \tag{8}$$

3. The calculation of the 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; Adali 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 function $f(\boldsymbol{v})$ with respect to variable $\boldsymbol{v}$ and its conjugate by

$$\begin{cases} \nabla_{\boldsymbol{v}} f = \dfrac{1}{3}\left(\nabla_{\boldsymbol{v}_a} f - \mathrm{j}\nabla_{\boldsymbol{v}_b} f - \mathrm{i}\nabla_{\boldsymbol{v}_c} f\right), \\ \nabla_{\boldsymbol{v}^*} f = \dfrac{1}{3}\left(\nabla_{\boldsymbol{v}_a} f + \mathrm{i}\nabla_{\boldsymbol{v}_b} f + \mathrm{j}\nabla_{\boldsymbol{v}_c} f\right). \end{cases} \tag{9}$$

Since trinions are commutative, the imaginary units i and j can be on any side of the real-valued gradients. The derivatives of some simple functions can be calculated, for example,

$$\frac{\partial v}{\partial v} = \frac{\partial v^*}{\partial v^*} = 1,\ \frac{\partial v}{\partial v^*} = \frac{\partial v^*}{\partial v} = \frac{1-\mathrm{i}+\mathrm{j}}{3}, \tag{10}$$

$$\begin{cases} \dfrac{\partial \mathrm{R}[\mathrm{tr}(\boldsymbol{V}\boldsymbol{W})]}{\partial \boldsymbol{V}} = \dfrac{1}{3}\boldsymbol{W}^{\mathrm{T}}, \\ \dfrac{\partial \mathrm{R}[\mathrm{tr}(\boldsymbol{W}\boldsymbol{V}^{\mathrm{H}})]}{\partial \boldsymbol{V}} = \dfrac{1}{3}\boldsymbol{W}^*, \\ \dfrac{\partial \mathrm{R}[\mathrm{tr}(\boldsymbol{V}\boldsymbol{W}\boldsymbol{V}^{\mathrm{H}})]}{\partial \boldsymbol{V}} = \dfrac{1}{3}\boldsymbol{V}^*(\boldsymbol{W}^* + \boldsymbol{W}^{\mathrm{T}}), \end{cases} \tag{11}$$

where $\boldsymbol{V}$ and $\boldsymbol{W}$ are trinion-valued matrices and $\mathrm{tr}(\cdot)$ denotes the matrix trace.

3 Trinion-valued filtering algorithms

3.1 Trinion-valued least mean squares 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

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

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

$$J(n) = |e(n)|^2. \tag{13}$$

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

$$\begin{aligned} \nabla_{\boldsymbol{w}^*} J(n) &= \frac{1}{3}\left[\nabla_{\boldsymbol{w}_a} J(n) + \mathrm{i}\nabla_{\boldsymbol{w}_b} J(n) + \mathrm{j}\nabla_{\boldsymbol{w}_c} J(n)\right] \\ &= \frac{2}{3}e(n)\boldsymbol{x}^*(n), \end{aligned} \tag{14}$$

yielding the following update equation for the weight vector:

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

where μ is the step size with the scale factor (2/3) absorbed into it. This LMS-like algorithm is termed 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

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

where $\boldsymbol{x}^{\mathrm{aug}}(n) = [\boldsymbol{x}(n)^{\mathrm{T}},\ \boldsymbol{x}^{\mathrm{i}}(n)^{\mathrm{T}},\ \boldsymbol{x}^{\mathrm{j}}(n)^{\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{w}^{\mathrm{aug}}(n) = [\boldsymbol{w}_1^{\mathrm{T}},\ \boldsymbol{w}_2^{\mathrm{T}},\ \boldsymbol{w}_3^{\mathrm{T}}]^{\mathrm{T}}$. Similarly, we have the following update equation for the augmented weight vector:

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

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 1, where the quaternion-valued LMS (QLMS) algorithm and the augmented QLMS (AQLMS) algorithm are based on the results in Barthélemy *et al.* (2014), Jiang *et al.* (2014), and Tao and Chang (2014). Clearly, the trinion model has a much lower complexity than the quaternion model.

Table 1 Number of real-valued operations (multiplication and addition) for each update of the weight vector

Algorithm	Number of multiplications	Number of additions
TLMS	$9L+3$	$9L$
QLMS	$16L+4$	$16L$
ATLMS	$27L+3$	$27L$
AQLMS	$64L+4$	$64L$

L: filter length. TLMS: trinion-valued least mean squares; ATLMS: augmented TLMS; QLMS: quaternion-valued least mean squares; AQLMS: augmented QLMS

3.2 Trinion-valued Kalman filter

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

$$\boldsymbol{x}_k = \boldsymbol{A}_k \boldsymbol{x}_{k-1} + \boldsymbol{B}_k \boldsymbol{u}_k + \boldsymbol{\omega}_k, \tag{18}$$

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

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

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k, \tag{19}$$

where $\boldsymbol{H}$ is the observation matrix and $\boldsymbol{v}_k$ the measurement noise. Both $\boldsymbol{\omega}_k$ and $\boldsymbol{v}_k$ are assumed to be zero-mean white-Gaussian, i.e., $\boldsymbol{\omega}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k)$ and $\boldsymbol{v}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_k)$. Here $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are the corresponding covariance matrices of the Gaussian processes. The 'a priori' and 'a posteriori' state estimates are respectively expressed as

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

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

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

$$\begin{aligned}
&E\{\|\boldsymbol{e}_{k|k}\|^2\} \\
=&\mathrm{R}\left\{\mathrm{tr}[\mathrm{cov}\left(\boldsymbol{x}_k - (\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}))\right)]\right\} \\
=&\mathrm{R}\left\{\mathrm{tr}[\mathrm{cov}\left((\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{v}_k)\right)]\right\},
\end{aligned} \tag{22}$$

where

$$\begin{aligned}
\boldsymbol{e}_{k|k} &= \boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k} \\
&= \boldsymbol{x}_k - [\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})].
\end{aligned} \tag{23}$$

Since the noise is independent of the states, we have

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

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

$$\begin{aligned}
E\{\|\boldsymbol{e}_{k|k}\|^2\} = \mathrm{R}\left[(\boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{H}\boldsymbol{P}_{k|k-1}\right. \\
\left. -\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{H}}\boldsymbol{K}_k^{\mathrm{H}} + \boldsymbol{K}_k\boldsymbol{S}_k\boldsymbol{K}_k^{\mathrm{H}})\right],
\end{aligned} \tag{25}$$

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

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

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

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

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

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

and subsequently we obtain the updated covariance matrix as follows:

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

This Kalman-like filter is termed the trinion-valued Kalman filter (TKF) and is summarized in Table 2.

Table 2 Trinion-valued Kalman filter

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

4 Simulation 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 (http://code.google.com/p/google-rec-csp/downloads/list). 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, in comparison with those of 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 initialized with all-zero filter coefficients. It can be observed that both augmented algorithms (AQLMS and ATLMS) have a similar larger 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 much lower computational complexity, as shown in Table 1.

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

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

It can be seen that the three sub-state vectors $\boldsymbol{x}_{ka}$, $\boldsymbol{x}_{kb}$, $\boldsymbol{x}_{kc}$ evolve dependently, and that the observation $\boldsymbol{z}_k$ is a linear mixture of them. The filtered results are plotted in Figs. 2–4. The errors in modulus before and after filtering are depicted in Fig. 5. We can see from the results that TKF can track the system state $\boldsymbol{x}_k$ effectively.

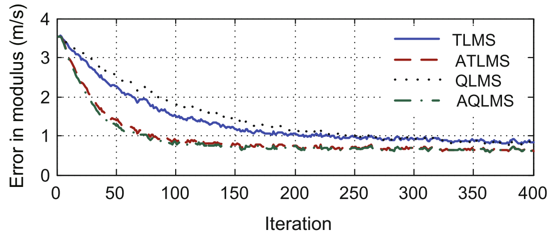

Fig. 1 Averaged learning curves

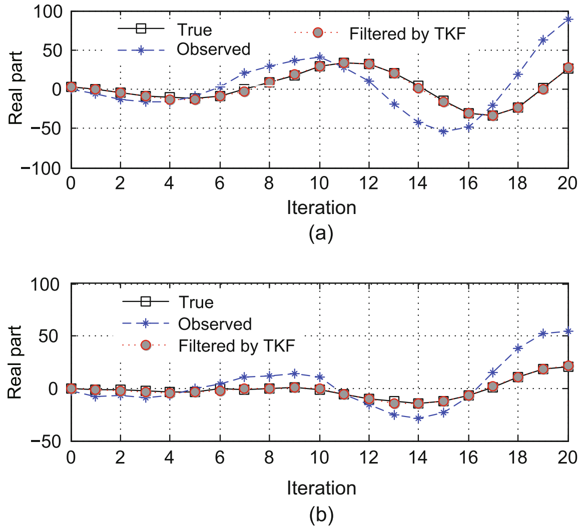

Fig. 2 Real part of the filtered results from TKF: (a) first element; (b) second element

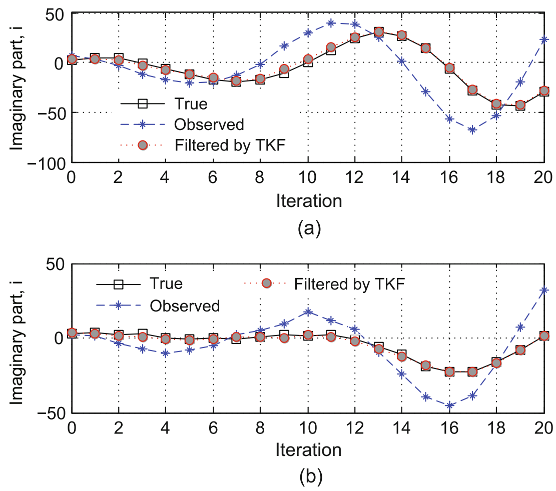

Fig. 3 Imaginary part (i) of the filtered results from TKF: (a) first element; (b) second element

5 Conclusions

A trinion-valued model for filtering and tracking of 3-D signals has been proposed. The corresponding algorithms were derived, including two LMS-type

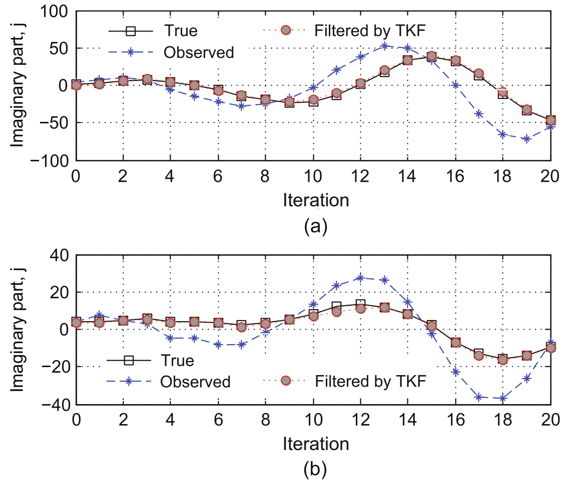

Fig. 4 Imaginary part (j) of the filtered results from TKF: (a) first element; (b) second element

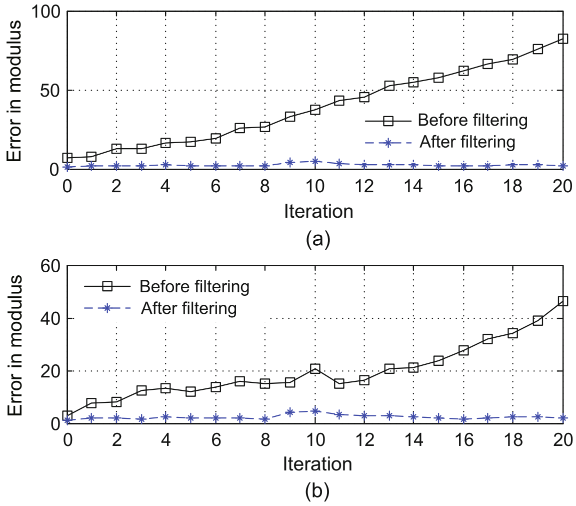

Fig. 5 Errors in modulus before and after filtering: (a) first element; (b) second element

algorithms (trinion-valued LMS and its augmented version) for adaptive filtering and a Kalman filtering algorithm for tracking. Simulation results show that the trinion model is a competitive candidate for 3-D signal processing with merits of reduced computational complexity (related to its compactness) and effective modeling of more complicated 3-D processes (related to its closure property).

References

Adali, T., Schreier, P.J., 2014. Optimization and estimation of complex-valued signals: theory and applications in filtering and blind source separation. *IEEE Signal Process. Mag.*, **31**(5):112-128. http://dx.doi.org/10.1109/MSP.2013.2287951

Allenby, R.B., 1991. Rings, Fields and Groups: Introduction to Abstract Algebra (Modular Mathematics Series). Elsevier Limited.

Assefa, D., Mansinha, L., Tiampo, K.F., *et al.*, 2011. The trinion Fourier transform of color images. *Signal Process.*, **91**(8):1887-1900. http://dx.doi.org/10.1016/j.sigpro.2011.02.011

Barthélemy, Q., Larue, A., Mars, J.I., 2014. About QLMS derivations. *IEEE Signal Process. Lett.*, **21**(2):240-243. http://dx.doi.org/10.1109/LSP.2014.2299066

Brandwood, D.H., 1983. A complex gradient operator and its application in adaptive array theory. *IEE Proc. H*, **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.

Gou, X., Liu, Z., Liu, W., *et al.*, 2015. Three-dimensional wind profile prediction with trinion-valued adaptive algorithms. Proc. IEEE Int. Conf. on Digital Signal Processing, p.566-569. http://dx.doi.org/10.1109/ICDSP.2015.7251937

Hawes, M., Liu, W., 2015. Design of fixed beamformers based on vector-sensor arrays. *Int. J. Antenn. Propag.*, **2015**:181937.1-181937.9. http://dx.doi.org/10.1155/2015/181937

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 Symp. on Combined Optical-Microwave Earth and Atmosphere Sensing, p.195-196. http://dx.doi.org/10.1109/COMEAS.1995.472367

Jahanchahi, C., Mandic, D.P., 2014. A class of quaternion Kalman filters. *IEEE Trans. Neur. Netw. Learn. Syst.*, **25**(3):533-544. http://dx.doi.org/10.1109/TNNLS.2013.2277540

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. 19th Int. Conf. on Digital Signal Processing, p.821-826. http://dx.doi.org/10.1109/ICDSP.2014.6900781

Jiang, M.D., Li, Y., Liu, W., 2016a. Properties of a general quaternion-valued gradient operator and its applications to signal processing. *Front. Inform. Technol. Electron. Eng.*, **17**(2):83-95. http://dx.doi.org/10.1631/FITEE.1500334

Jiang, M.D., Liu, W., Li, Y., 2016b. Adaptive beamforming for vector-sensor arrays based on a reweighted zero-attracting quaternion-valued LMS algorithm. *IEEE Trans. Circ. Syst. II*, **63**(3):274-278. http://dx.doi.org/10.1109/TCSII.2015.2482464

Kantor, I.L., Solodovnikov, A.S., 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 Process.*, **84**(7):1177-1199. http://dx.doi.org/10.1016/j.sigpro.2004.04.001

Le Bihan, N., Miron, S., Mars, J.I., 2007. MUSIC algorithm for vector-sensors array using biquaternions. *IEEE Trans. Signal Process.*, **55**(9):4523-4533. http://dx.doi.org/10.1109/TSP.2007.896067

Li, T.C., Villarrubia, G., Sun, S.D., *et al.*, 2015. Resampling methods for particle filtering: identical distribution, a new method, and comparable study. *Front. Inform. Technol. Electron. Eng.*, **16**(11):969-984. http://dx.doi.org/10.1631/FITEE.1500199

Liu, H., Zhou, Y.L., Gu, Z.P., 2014. Inertial measurement unit-camera calibration based on incomplete inertial sensor information. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **15**(11):999-1008. http://dx.doi.org/10.1631/jzus.C1400038

Liu, W., 2014. Antenna array signal processing for quaternion-valued wireless communication system. Proc. IEEE Benjamin Franklin Symp. on Microwave and Antenna Sub-systems, p.1-3.

Miron, S., Le Bihan, N., Mars, J.I., 2006. Quaternion-MUSIC for vector-sensor array processing. *IEEE Trans. Signal Process.*, **54**(4):1218-1229. http://dx.doi.org/10.1109/TSP.2006.870630

Parfieniuk, M., Petrovsky, A., 2010. Inherently lossless structures for eight- and six-channel linear-phase paraunitary filter banks based on quaternion multipliers. *Signal Process.*, **90**(6):1755-1767. http://dx.doi.org/10.1016/j.sigpro.2010.01.008

Pei, S.C., Cheng, C.M., 1999. Color image processing by using binary quaternion-moment-preserving thresholding technique. *IEEE Trans. Image Process.*, **8**(5):614-628. http://dx.doi.org/10.1109/83.760310

Pei, S.C., Chang, J.H., Ding, J.J., 2004. Commutative reduced biquaternions and their Fourier transform for signal and image processing applications. *IEEE Trans. Signal Process.*, **52**(7):2012-2031. http://dx.doi.org/10.1109/TSP.2004.828901

Sangwine, S.J., Ell, T.A., Blackledge, J.M., *et al.*, 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. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, p.3956-3960. http://dx.doi.org/10.1109/ICASSP.2015.7178713

Tao, J.W., 2013. Performance analysis for interference and noise canceller based on hypercomplex and spatio-temporal-polarisation processes. *IET Radar Sonar Navig.*, **7**(3):277-286. http://dx.doi.org/10.1049/iet-rsn.2012.0151

Tao, J.W., Chang, W.X., 2014. Adaptive beamforming based on complex quaternion processes. *Math. Prob. Eng.*, **2014**:291249.1-291249.10. http://dx.doi.org/10.1155/2014/291249

Ward, J.P., 1997. Quaternions and Cayley Numbers: Algebra and Applications. Springer, the Netherlands. http://dx.doi.org/10.1007/978-94-011-5768-1

Zetterberg, L., Brandstrom, H., 1977. Codes for combined phase and amplitude modulated signals in a four-dimensional space. *IEEE Trans. Commun.*, **25**(9):943-950. http://dx.doi.org/10.1109/TCOM.1977.1093927

Zhang, X.R., Liu, W., Xu, Y.G., *et al.*, 2014. Quaternion-valued robust adaptive beamformer for electromagnetic vector-sensor arrays with worst-case constraint. *Signal Process.*, **104**:274-283. http://dx.doi.org/10.1016/j.sigpro.2014.04.006

Appendix A: Calculation of the gradient in Eq. (14)

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

$$\begin{aligned} J =& (d_a - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_a + \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_c + \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_b)^2 \\ &+ (d_b - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_b - \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_a + \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_c)^2 \\ &+ (d_c - \boldsymbol{w}_a^{\mathrm{T}}\boldsymbol{x}_c - \boldsymbol{w}_b^{\mathrm{T}}\boldsymbol{x}_b - \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}_a)^2, \end{aligned} \tag{A1}$$

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.,

$$\begin{aligned} \nabla_{\boldsymbol{w}_a} J =& 2\big[(\boldsymbol{x}_a\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_b^{\mathrm{T}} + \boldsymbol{x}_c\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_a \\ &+ (\boldsymbol{x}_b\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_c\boldsymbol{x}_b^{\mathrm{T}} - \boldsymbol{x}_a\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_b \\ &+ (\boldsymbol{x}_c\boldsymbol{x}_a^{\mathrm{T}} - \boldsymbol{x}_a\boldsymbol{x}_b^{\mathrm{T}} - \boldsymbol{x}_b\boldsymbol{x}_c^{\mathrm{T}})\boldsymbol{w}_c \\ &- (d_a\boldsymbol{x}_a + d_b\boldsymbol{x}_b + d_c\boldsymbol{x}_c)\big], \end{aligned} \tag{A2}$$

$$\begin{aligned} \nabla_{\boldsymbol{w}_b} J =& 2\big[(\boldsymbol{x}_a\boldsymbol{x}_b^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_c^{\mathrm{T}} - \boldsymbol{x}_c\boldsymbol{x}_a^{\mathrm{T}})\boldsymbol{w}_a \\ &+ (\boldsymbol{x}_c\boldsymbol{x}_c^{\mathrm{T}} + \boldsymbol{x}_a\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_b^{\mathrm{T}})\boldsymbol{w}_b \\ &+ (\boldsymbol{x}_c\boldsymbol{x}_b^{\mathrm{T}} - \boldsymbol{x}_a\boldsymbol{x}_c^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_a^{\mathrm{T}})\boldsymbol{w}_c \\ &+ (d_a\boldsymbol{x}_c - d_b\boldsymbol{x}_a - d_c\boldsymbol{x}_b)\big], \end{aligned} \tag{A3}$$

$$\begin{aligned} \nabla_{\boldsymbol{w}_c} J =& 2\big[(\boldsymbol{x}_a\boldsymbol{x}_c^{\mathrm{T}} - \boldsymbol{x}_b\boldsymbol{x}_a^{\mathrm{T}} - \boldsymbol{x}_c\boldsymbol{x}_b^{\mathrm{T}})\boldsymbol{w}_a \\ &+ (\boldsymbol{x}_b\boldsymbol{x}_c^{\mathrm{T}} - \boldsymbol{x}_c\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_a\boldsymbol{x}_b^{\mathrm{T}})\boldsymbol{w}_b \\ &+ (\boldsymbol{x}_a\boldsymbol{x}_a^{\mathrm{T}} + \boldsymbol{x}_c\boldsymbol{x}_c^{\mathrm{T}} + \boldsymbol{x}_b\boldsymbol{x}_b^{\mathrm{T}})\boldsymbol{w}_c \\ &+ (d_a\boldsymbol{x}_b + d_b\boldsymbol{x}_c - d_c\boldsymbol{x}_a)\big]. \end{aligned} \tag{A4}$$

Finally, the gradient of $J(n)$ is obtained by merging Eqs. (A2)–(A4) into Eq. (14):

$$\nabla_{\boldsymbol{w}^*} J(n) = \frac{2}{3} e(n)\boldsymbol{x}^*(n). \tag{A5}$$

Appendix B: Calculation of the Kalman gain matrix

We know from Eqs. (11) and (26) that

$$-\frac{1}{3}\boldsymbol{P}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{H}} - \frac{1}{3}\boldsymbol{P}_{k|k-1}^{*}\boldsymbol{H}^{\mathrm{T}} + \frac{1}{3}\boldsymbol{K}_k(\boldsymbol{S}_k^{*} + \boldsymbol{S}_k^{\mathrm{T}}) = \boldsymbol{0}. \tag{B1}$$

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

$$\boldsymbol{S}_k^{*} = \boldsymbol{S}_k^{\mathrm{T}},\ \boldsymbol{P}_{k|k-1}^{*} = \boldsymbol{P}_{k|k-1}^{\mathrm{T}}, \tag{B2}$$

we have

$$-\frac{1}{3}\boldsymbol{P}_{k|k-1}^{*}\boldsymbol{H}^{\mathrm{H}} - \frac{1}{3}\boldsymbol{P}_{k|k-1}^{*}\boldsymbol{H}^{\mathrm{T}} + \frac{2}{3}\boldsymbol{K}_k\boldsymbol{S}_k^{*} = \boldsymbol{0}, \tag{B3}$$

which yields Eq. (27).