A Normalized Adaptive Filtering Algorithm Based on Geometric Algebra

RUI WANG\textsuperscript{1}, (Senior Member, IEEE), MEIXIANG LIANG\textsuperscript{1}, YINMEI HE\textsuperscript{1}, XIANGYANG WANG\textsuperscript{1}, AND WENMING CAO\textsuperscript{2}, (Member, IEEE)

\textsuperscript{1}Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Joint International Research Laboratory of Specialty Fiber Optics and Advanced Communication, School of Communication and Information Engineering, Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China
\textsuperscript{2}College of Information Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding author: Wenming Cao (wmcao@szu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61771299 and Grant 61771322.

\begin{abstract}
In this paper, we extend the original Normalized Least Mean Fourth (NLMF) and Normalized Least Mean Square (NLMS) adaptive filtering algorithms into Geometric Algebra (GA) space to enable them to process multidimensional signals. We redefine the cost functions and propose the GA based NLMF and NLMS algorithms (GA-NLMF & GA-NLMS). We take full advantage of the ability of GA to represent multidimensional signals in GA space. GA-NLMS minimizes the cost function of the normalized mean square of the error signal, and remain stable as the input signal of the filter increases. GA-NLMS has fast convergence rate but higher steady-state error. The GA-NLMF algorithm minimizes the cost function of the normalized mean fourth of the error signal. Simulation results show that our proposed GA-NLMS adaptive filtering algorithm outperforms original NLMS algorithm in terms of convergence rate and steady-state error, and GA-NLMF outperforms both NLMF and GA-NLMS algorithms. GA-NLMF has faster convergence rate and lower steady state error, which is proved in the experiments.
\end{abstract}

\begin{IEEEkeywords}
Geometric algebra, normalized least mean fourth, normalized least mean square, adaptive filters.
\end{IEEEkeywords}

\section{I. INTRODUCTION}

Adaptive filters (AFs) are usually described via standard vector calculus and linear algebra [1]. Almost all existing filter algorithms iterate by minimizing the mean square error cost function [2]–[4], the least mean-square (LMS) algorithm is the most common filter algorithm [2]. However, the performance of the LMS algorithm is not optimal, the main drawback of the LMS algorithm is that it is sensitive to the input signal, the NLMS algorithm is proposed to solve it by normalizing with the power of the input signal [5]. The NLMS algorithm is most popular due to its simplicity, but the stability is controlled by the fixed step size, since the step size affects convergence rate and mean square error, C.Paleologu propose a variable step-size NLMS (VSS-NLMS) algorithm [6], thus it solves the conflict between convergence rate and low excess mean square error associated with the conventional NLMS algorithm [6]–[10]. Another type of variable step-size algorithm is the regularized NLMS algorithm, Choi present a robust regularized NLMS (RR-NLMS) algorithm [7], which updates regularization parameter adaptively by using normalized gradient, the convergence rate has been improved. Although there are many improved algorithms based on NLMS algorithm and the convergence rate has been improved to some extent, the convergence rate is still relatively slow. Compared with NLMS algorithm, the filter algorithm based on high-order statistical has better performance. For example, least mean fourth (LMF) algorithm has a good performance than NLMS in terms of convergence rate and steady state error [11], but the LMF algorithm cannot guarantee stability when the input power of the adaptive filter increases. Therefore, the NLMF is proposed which remains stable relative to LMF algorithm [12], the NLMF algorithm has also been used successfully in many applications. In order to apply the NLMF algorithm to the problem of channel equalization with co-channel interference and additive white Gaussian noise, Zerguine et al. [13] propose the normalized LMF (XE-NLMF), but the calculation cost is relatively increased a little, and XE-NLMF algorithm is mainly for real-valued signal. Thus, Faiz and Zerguine and [14] propose the normalized
sign regressor least mean fourth (NSRLMF) adaptive algorithm for complex-valued and real-valued signals. However, NLMF-type algorithms have not been used to process multidimensional signals, to solve this problem, we introduce the concept of geometric algebra (GA), which represents the multidimensional signal as a multivector in GA space.

In GA space, geometric algebra provides an efficient computing framework without using coordinate information, it simplifies the complexity of computation [15]–[19]. Recently, the LMS filter algorithm based on GA theory is proposed, GA-LMS algorithm has been applied for 3-D point-clouds registration problem and the 6 degrees of freedom (6-DOF) recovery transformation [20], [21]. GA-LMS algorithm has been widely used in signal processing, however, the convergence rate is slow. Compared with the GA-LMS algorithm, the GA-LMK [22] algorithm has improved the convergence performance and steady-state error, but it is sensitive to the input signal. Thus, the NLMF and NLMS solve the problem of sensitivity to input signals, they provide significant performance improvement in terms of the convergence rate, but they are not very good at processing multi-dimensional signals, which is the focus of this paper.

This contribution of this paper is as follows: First, taking the advantage of geometric algebra theory, GA-NLMS and GA-NLMF algorithm represent a multidimensional signal as a multivector. Compared with the original NLMS and NLMF algorithm, they process multidimensional signal in a holistic way. Second, the GA-NLMS is proposed for the processing of multidimensional signal, GA-NLMS algorithm minimizes the cost function based on the mean square normalized estimation error, and has a certain improvement in the convergence rate, but it has higher steady-state error. Therefore we also propose a normalized least mean fourth filtering algorithm based on geometric algebra, we prove that the proposed GA-NLMS algorithm has a great improvement in steady-state error and convergence rate than GA-NLMS algorithm and some existing algorithms, the GA-NLMS adaptive filtering algorithm may eventually become a powerful tool for multi-dimensional signal processing.

This paper is organized as follows: Section II introduces the basis of geometric algebra and geometric calculus, and the original NLMS and NLMS algorithm. Section III introduces GA-NLMS algorithm and GA-NLMF algorithm. Simulation results are shown in section IV to analyze the performance of GA-NLMF and GA-NLMS algorithm. Finally, section V gives the conclusion.

II. GEOMETRIC ALGEBRA (GA)

A. THE BASIC OF GEOMETRIC ALGEBRA

In this section, we review the basis of the geometric algebra (GA), let $G_n$ denote the geometric algebra of the $n$-dimensions space, and the $G_n$ is a geometric extension of the $R_n$, which enables algebraic representation of magnitude and orientations [23]–[26]. Geometric product is the most important operation form of GA space, the geometric product includes inner product and outer product. Consider vector $a$ and $b$ in $R_n$, the inner product of vectors $a$ and $b$ can be defined as

$$a \cdot b = |a||b| \cos \phi$$

where $|\cdot|$ denotes the magnitude of the vector of $a$ and $b$, $\phi$ is the angle between $a$ and $b$, since formula (1) is a scalar, the inner product is commutative, i.e., $a \cdot b = b \cdot a$.

The outer product $a \wedge b$ is the product of the exterior algebra introduced by Grassmann’s Ausdehnungslehre [20], [27]–[29]. It is a parallelogram surface represented by two directed segments, such a surface can be interpreted as hyperplane when vector $a$ is swept on the direction determined by vector $b$, the outer product is defined as follows

$$a \wedge b = I_{a,b} |a||b| \sin \phi$$

where $I_{a,b}$ is the unit bivector that denotes the orientation of the hyperplane $a \wedge b$, since formula (2) is a vector and it is directivity, the inner product is noncommutative, i.e., $a \wedge b = -b \wedge a$.

Finally, the geometric product of the vector $a$ and $b$ is denoted by $ab$ and defined as follows

$$ab = a \cdot b + a \wedge b$$

due to the outer product is noncommutative, the geometric product also is noncommutative, that is $ab \neq ba$.

In GA space, each orthogonal basis in $R_n$ generates $2^n$ members via geometric product [30], [31], we can denote a $n$-dimensions GA space by an orthonormal basis of vectors $e_1, e_2, \cdots e_n$, then, we can obtain the basis of the GA space as follows

$$\{1, e_1, e_1e_2, \cdots, e_1e_2\cdots e_n\}$$

For the $R_3$ case, $G_3$ has the $2^3 = 8$ elements, with the basis of $\{1, e_1, e_2, e_3, e_{12}, e_{23}, e_{13}, I\}$, i.e., one scalar, three orthogonal vector $e_i$, three bivectors $e_{ij} = e_i e_j$ and one trivector $I = e_1 e_2 e_3$. In addition, a multivector $A$ can be defined as follows [32]

$$A = \langle A \rangle_0 + \langle A \rangle_1 + \langle A \rangle_2 + \cdots = \sum_k \langle A \rangle_k$$

which is made up of its k-vectors (or k-grades) $\langle A \rangle_k$ and $k=0$ (scalars), $k=1$ (vectors), $k=2$ (bivectors), $k=3$ (trivectors), and so on. GA theory has the ability to combine scalars, vectors, and super-complex quantities in a single element (multivector) [33]–[36]. Another example is provided to denote the reverse of a multivector $A$ and defined as follows

$$\tilde{A} = \sum_{k=0}^n (-1)^{(k+1)/2} \langle A \rangle_k$$

For example, for $A = \langle A \rangle_0 + \langle A \rangle_1 + \langle A \rangle_2$, then the reverse of $A$ is $\tilde{A} = \langle A \rangle_0 + \langle A \rangle_1 - (\langle A \rangle_2)$. Noting that the 0-vector of the multivector is not affected by reversion, so $\langle A \rangle_0 = \langle \tilde{A} \rangle_0$, in addition, the scalar product $\langle \cdot \rangle$ is defined as follows

$$A \ast B = \langle AB \rangle_0$$
where the scalar product is the scalar part (0-vector) of the geometric multiplication between $A$ and $B$.

Besides, an array of multivectors is made up of the general multivector. For $G_3$ case, given $M$ multivectors $\{U_1, U_2, \cdots, U_M\}$, the array can be represented as follows [30]

$$
\mathbf{u} = \begin{bmatrix}
U_1 \\
\vdots \\
U_M
\end{bmatrix} = \begin{bmatrix}
u(1, 0) + u(1, 1)e_1 + \cdots + u(1, 7)I \\
\vdots \\
u(M, 0) + u(M, 1)e_1 + \cdots + u(M, 7)I
\end{bmatrix}
$$

(8)

This work adopts the notion of matrix of multivectors: the elements of the matrix is made up of the multivector-valued. Similarly, we can give a multivectors $\{W_1, W_2, \cdots W_M\}$, the definition of the multivectors $\mathbf{w}$ as follows

$$
\mathbf{w} = \begin{bmatrix}
W_1 \\
\vdots \\
W_M
\end{bmatrix} = \begin{bmatrix}
w(1, 0) + w(1, 1)e_1 + \cdots + w(1, 7)I \\
\vdots \\
w(M, 0) + w(M, 1)e_1 + \cdots + w(M, 7)I
\end{bmatrix}
$$

(9)

Then, we can get the array product between them, the definition of the array product as follows

$$
\mathbf{u}^T \mathbf{w} = U_1 W_1 + U_2 W_2 + \cdots + U_M W_M = \sum_{i=1}^{M} U_i W_i
$$

(10)

where the terms $U_i W_i$ is the geometric product, and the results of $\mathbf{u}^T \mathbf{w}$ is a multivector. Similarly, the reverse transpose array is the extension of the reverse operation of multivectors to include arrays of multivectors. Therefore, we can get the following operation

$$
\mathbf{u}^* \mathbf{w} = \bar{U}_1 W_1 + \bar{U}_2 W_2 + \cdots + \bar{U}_M W_M = \sum_{i=1}^{M} \bar{U}_i W_i
$$

(11)

where $\mathbf{u}^* = \{\bar{U}_1, \bar{U}_2, \cdots, \bar{U}_M\}$, it is the reverse version of the $\mathbf{u}$. In GA space, we know that the array product between $\mathbf{u}$ and $\mathbf{u}^*$ is represented by $\|\mathbf{u}\|^2 = \mathbf{u}^* \mathbf{u}$, it is a multivector. In addition, the distance is given by the magnitude of a multivector can be defined the scalar product, the form is as follows

$$
\|\mathbf{u}\|^2 = \mathbf{u} \ast \bar{\mathbf{u}} = \bar{\mathbf{u}} \ast \mathbf{u} = \sum_{k} |\langle \mathbf{u} \rangle_k|^2
$$

(12)

### B. GENERAL COST FUNCTION IN GA SPACE

Following the guidelines in [30], we can formulate the minimization problem in geometric algebra and the cost function as follows

$$
J(D, P_k, X, Q_k) = |D - \sum_{k=1}^{M} P_k X Q_k|
$$

(13)

where $D, P_k, X, Q_k$ are all general multivectors, and $M$ is the number of the taps in the adaptive filter. By setting different values for the parameters in this formula, the cost functions can be obtained in different cases. In this paper, we select $P_k = U_k X = 1, Q_k = W_k P_k, Q_k$ are the general multivectors, then we can obtain the estimation error $E = D - \mathbf{u}^* \mathbf{w}$, where $\mathbf{u}^* \mathbf{w}$ is the array product.

### C. RELATED ALGORITHM

1) **NORMALIZED LEAST MEAN SQUARE ALGORITHM**

This section introduces a normalized least mean square (NLMS) algorithm [5]. Its cost function is based on the normalized mean square estimated error $e$, and its estimated error form is as follows

$$
e_k = a_k - h_k^T x_k
$$

(14)

where $x_k$ denote the input vector at the time $k$, and $a_k$ denotes the output vector at the time $k$ and $h_k$ is the weight vector of the adaptive filter, $e_k$ is the error between the output signal and the expected signal.

We can get the cost function from [5], the proposed algorithm is based on minimizing the cost function as follows

$$
J(h) = E \left( \frac{e_k}{\|x_k\|} \right)^2
$$

(15)

By taking the derivative of the formula(15), the gradient of $J(h)$ with respect to $h$ is

$$
\frac{\partial J(h)}{\partial h} = -2 \left( \frac{e_k x_k}{\|x_k\|^2} \right)
$$

(16)

Then the corresponding gradient algorithm is $h_{k+1} = h_k + \mu \frac{e_k x_k}{\|x_k\|^2}$, this is the proposed NLMS algorithm. We know that the normalization of the gradient algorithm is a division of the weight vector update term of the regular LMS algorithm by the mean square of the input norm. So the normalized weight vector update term remains bound as $\|x_k\|$ increases. The algorithm stability does not depend on the input signal power of the adaptive filter.

2) **NORMALIZED LEAST MEAN FOURTH ALGORITHM**

This section introduces a normalized least Mean Fourth (NLMF) algorithm [12], the derivation process is similar to NLMS algorithm. Its cost function is based on the normalized mean fourth estimated error $e$. From [12], the NLMF algorithm is based on minimizing the cost function as follows

$$
J(h) = E \left( \frac{e_k^4}{\|x_k\|} \right)
$$

(17)

By taking the derivative of the formula (17), the gradient of $J(h)$ with respect to $h$ is

$$
\frac{\partial J(h)}{\partial h} = -4 \left( \frac{e_k^3 x_k}{\|x_k\|^4} \right)
$$

(18)

Then the corresponding gradient algorithm is $h_{k+1} = h_k + \mu \frac{e_k^3 x_k}{\|x_k\|^4}$, this is the proposed NLMF algorithm. In details, unlike LMF algorithm, the normalization of the gradient algorithm is the weight vector update term of the LMF algorithm divided by the input norm to the fourth power.
III. THE PROPOSED ALGORITHM

A. NORMALIZED LEAST SQUARE ALGORITHM BASED ON GEOMETRIC ALGEBRA

In this section, we derive the GA-NLMS algorithm by using Geometric algebra and Geometric calculus (GC) theory \[37\]. The cost function is based on the mean square normalized estimation error \( E \), we need to minimize the cost function, it is necessary for us to update the rules by computing the derivatives of multivector in the GA space. Thus, at instant \( i \), yield an instantaneous cost function \( J(i) \) \[38\]–\[40\], so that

\[
J(w_{i-1}) = \mathbb{E}\left\{ \left| E(i) \right|^2 \right\} \tag{19}
\]

where \( E(i) = D(i) - \tilde{D}(i) \), \( \tilde{D}(i) = u_i \ast w_{i-1} \) is an estimate of \( D(i) \), \( u(i) \) is the input vector signal, and \( \tilde{D}(i) \) is the output vector signal.

Given \( E(i) \) and \( w_{i-1} \), we can update the array of multivectors \( w \) by a recursive rule of the form \[31\]

\[
w_i = w_{i-1} + \mu h
\]

where \( \mu \) is the adaptive filter step size, and \( h \) is an array of the multivectors related to the estimation error \( E(i) \). The design direction of the adaptive filter adopted in this work is opposite to the instantaneous gradient direction of the instantaneous steepest-descent rule \[1\], \[39\], \[40\]. So we can obtain \( h = -G\partial_w J(w_{i-1}) \), yielding the form of GA-based adaptive filter rules.

\[
w_i = w_{i-1} - \mu G\partial_w J(w_{i-1}) \tag{21}
\]

where \( G \) is a matrix with multivector entries, choosing a suitable can lead to different types of adaptive algorithm. In this work, we setting \( G \) as an identity matrix.

In order to obtain GA-NLMS adaptive rule, we need to calculate a multivector derivative from the equation (21). we know that the differential operator \( \partial_w \) has the algebra properties of the multivector in GA space \[40\]. Therefore the gradient \( \partial_w J(w_{i-1}) \) can be seen as the geometric multiplication between \( \partial_w \) and \( J(w_{i-1}) \).

In GA space, we know that any multivector \( A \) can be decomposed into blades via \[31\]

\[
A = \sum_{k=0}^{2^n-1} \lambda_k (\lambda^k \ast A) = \sum_{k=0}^{2^n-1} \lambda_k \tilde{A}^k = \sum_{k=0}^{2^n-1} \lambda_k A^k \tag{22}
\]

where \( A_k \) is a scalar, and \( \lambda, \lambda^k, k = 0, 1, 2, \ldots 2^n - 1 \) are the reciprocal basis of \( G_n \), we know that the following relation holds for the reciprocal bases: \( \lambda^i \cdot \lambda_j = \delta_{ij} \), where \( \delta_{ij} = 1, i = j \), and \( \delta_{ij} = 0, i \neq j \), similarly, it is easy to know that \( \lambda_k \cdot \tilde{\lambda}_k \) comply with relation above(they are reciprocal basis), then we can use the formula(22) to decompose \( \partial_w \) into blades \[31\]

\[
\partial_w = \sum_{k=0}^{2^n-1} \lambda_k \langle \lambda_k \partial_w \rangle = \sum_{k=0}^{2^n-1} \lambda_k \partial_w \langle k \rangle \tag{23}
\]

where \( \langle k \rangle \) is the basis of \( G_n \) and \( \partial_w \langle k \rangle \) denotes the general derivative from standard calculus, which only affects blades.

Noticing that the formula (19) can be rewritten in terms of the scalar product as follows

\[
J(w_{i-1}) = \mathbb{E}\left\{ \left| \frac{E(i)}{||u_i||^2} \right|^2 \right\} = \mathbb{E}\left\{ \frac{E(i) \ast \tilde{E}(i)}{||u_i||^2} \right\} \tag{24}
\]

where \( u_i \) is the arrays of multivector, its square power is a multivector, the denominator is the norm of a multivector vector, which is a scalar, \( E(i) \) denotes the error between the estimate signal and the actual signal, and it can be decompose in term of its blades via (22)

\[
E(i) = \sum_{s=0}^{2^n-1} \lambda_s E_s, \tilde{E}(i) = \sum_{s=0}^{2^n-1} E_s \tilde{E}_s \tag{25}
\]

In order to derive the GA-NLMS adaptive rule further ahead, we need to calculate the \( \partial_w J(w_{i-1}) \), we replace the expectation of partial derivatives with the instantaneous estimate, then, simplify it as

\[
\partial_w J(w_{i-1}) = \partial_w \mathbb{E}\left\{ \left| \frac{E(i)}{||u_i||^2} \right|^2 \right\} = \partial_w \mathbb{E}\left\{ \frac{E(i) \ast \tilde{E}(i)}{||u_i||^2} \right\} = \partial_w \mathbb{E}\left\{ \frac{E(i) \ast \tilde{E}(i)}{||u_i||^2} \right\} \tag{26}
\]

According to (25), we can obtain the following equation

\[
E(i) \ast \tilde{E}(i) = \sum_{s=0}^{2^n-1} \lambda_s E_s \ast \sum_{s=0}^{2^n-1} E_s \tilde{E}_s = \sum_{s=0}^{2^n-1} E_s^2 (\lambda_s \ast \tilde{E}_s) = \sum_{s=0}^{2^n-1} E_s^2 \tag{27}
\]

Therefore, plugging (23) and (27) into (26) results in

\[
\partial_w J(w_{i-1}) = \partial_w \left\{ \frac{E(i)^2}{||u_i||^2} \right\} = \sum_{l=0}^{2^n-1} \lambda_l \partial_{\lambda_l} \left( \frac{E^2}{||u_i||^2} \right) = \sum_{l=0}^{2^n-1} \lambda_l \partial_{\lambda_l} E^2 = \sum_{l=0}^{2^n-1} \lambda_l \partial_{\lambda_l} E^2 \tag{28}
\]

We know that \( E(i) = D(i) - \tilde{D}(i) \), thus the estimation error can be defined as

\[
E_s = D_s - \tilde{D}_s, \tilde{D}_s = \langle \tilde{D}(i) \rangle_s = \langle u_i \ast w_{i-1} \rangle_s \tag{29}
\]
Thus, \( \partial_{w,j} E^2 \) can be obtained by using the estimation error in (29) as follows
\[
\partial_{w,j} E^2 = 2E_s(\partial_{w,j} E_s) = 2E_s(\partial_{w,j}(D_i - \hat{D}_i)) = -2E_s \partial_{w,j} \hat{D}_i
\]
(30)
since \( D_i \) does not depend on the weight array \( w \), so the \( \partial_{w,j} D_i \) is obtained by decomposing \( \hat{D}_i \) into its blades [31]
\[
\hat{D}(i) = u^*_i w_{i-1} = \sum_{s=0}^{2^n-1} \lambda_s(\hat{\lambda}_s\{u^*_i w_{i-1}\})
\]
(31)
which requires to perform the decomposition on \( u^*_i \) and \( w_{i-1} \).
Thus employing (5) once again, \( u^*_i \) and \( w_{i-1} \) can be decomposed into blades as follows
\[
u^*_i = \sum_{q=0}^{2^n-1} u^T_q \hat{\lambda}_q = \sum_{q=0}^{2^n-1} u^T_q \lambda_q
\]
\[
w_{i-1} = \sum_{t=0}^{2^n-1} \lambda_t w_{i-1,t} = \sum_{t=0}^{2^n-1} \lambda_t w_t
\]
(32)
where \( u^T_q \) and \( w_{i-1} \) are respectively \( 1 \times M \) and \( M \times 1 \) arrays with real entries, then plugging (32) into (31), we obtain
\[
\hat{D}(i) = u^*_i w_{i-1} = \sum_{s=0}^{2^n-1} \lambda_s(\hat{\lambda}_s\{u^*_i w_{i-1}\}) = \sum_{s=0}^{2^n-1} \lambda_s\{\hat{\lambda}_s u^*_i w_{i-1}\}
\]
\[
= \sum_{s=0}^{2^n-1} \lambda_s\{\hat{\lambda}_s u^*_i w_{i-1}\} = \sum_{s=0}^{2^n-1} \lambda_s\{\hat{\lambda}_s u^*_i w_{i-1}\}
\]
\[
= \sum_{s=0}^{2^n-1} \lambda_s\{\hat{\lambda}_s u^*_i w_{i-1}\} = \sum_{s=0}^{2^n-1} \lambda_s\{\hat{\lambda}_s u^*_i w_{i-1}\}
\]
(33)
From (33), we can obtain the following equation
\[
\hat{D}_i = \sum_{q,t=0}^{2^n-1} (\hat{\lambda}_s \hat{\lambda}_q \lambda_t)\{u^*_q w_t\}
\]
(34)
Then plugging (34) into (30), we can obtain
\[
\partial_{w,i} E^2 = 2E_s(\partial_{w,j} E_s) = -2E_s \partial_{w,j} \hat{D}_i
\]
\[
= -2E_s \sum_{q,t=0}^{2^n-1} (\hat{\lambda}_s \hat{\lambda}_q \lambda_t)\{u^*_q w_t\}
\]
\[
= -2E_s \sum_{q,t=0}^{2^n-1} (\hat{\lambda}_s \hat{\lambda}_q \lambda_t)\{u^*_q w_t\}
\]
\[
= -2E_s \sum_{q,t=0}^{2^n-1} (\hat{\lambda}_s \hat{\lambda}_q \lambda_t)\{u^*_q w_t\}
\]
(35)
Noticing that \( \partial_{w,i} w_t \) will be different from zero only when \( l = t \), thus, adopting the Kronecker delta function [31], \( \partial_{w,i} E^2 = \delta_{i,i}. \) Finally, substituting (35) into (28), the stochastic gradient is obtained
\[
\partial_w J(w_{i-1}) = \frac{-2 \sum_{l=0}^{2^n-1} \lambda_l \partial_{w,l} E^2_s}{\{\|u_i\|^2\}^2}
\]
\[
= \frac{-2 \sum_{l=0}^{2^n-1} \lambda_l \{\|u_i\|^2\}^2}{\{\|u_i\|^2\}^2}
\]
\[
= \frac{-2 \sum_{l=0}^{2^n-1} \lambda_l \{\|u_i\|^2\}^2}{\{\|u_i\|^2\}^2}
\]
\[
= \frac{-2 \sum_{l=0}^{2^n-1} \lambda_l \{\|u_i\|^2\}^2}{\{\|u_i\|^2\}^2}
\]
\[
= \frac{-2 \sum_{l=0}^{2^n-1} \lambda_l \{\|u_i\|^2\}^2}{\{\|u_i\|^2\}^2}
\]
(36)
According to the GAAFs literature, we can set the \( G \) equal to the identify matrix in (21), then yield the GA-NLMS update rule.
\[
w_i = w_{i-1} + \mu u_i E(i)
\]
\[
\|u_i\|^2
\]
where \( \mu \) is the step size of the adaptive filter algorithm.

**B. NORMALIZED LEAST MEAN FOURTH ALGORITHM BASED ON GEOMETRIC ALGEBRA**

In this section, we derive the GA-NLMF algorithm by using geometric algebra(GA) and geometric calculus(GC) theory, the cost function is based on the fourth power of the normalized estimation error \( E \), we need to minimize the cost function, similar to the derivation of the GA-NLMS, we can obtain the rule for updating the GA-NLMF, thus the cost function \( J(i) \) [39], [40], so that
\[
J(w_{i-1}) = \mathbb{E}\left\{|E(i)|^4\|u_i\|^4\right\}
\]
(38)
Noticing that the formula (38) can be rewritten in terms of the scalar product as follows
\[
J(w_{i-1}) = \mathbb{E}\left\{|E(i)|^4\|u_i\|^4\right\} = \mathbb{E}\left\{E(i) \ast \tilde{E}(i) \ast E(i) \ast \tilde{E}(i)\right\}
\]
(39)
In order to derive the GA-NLMF adaptive rule further ahead, we need to calculate the $\partial w / \partial J(w_{l-1})$, we replace the expectation of partial derivatives with the instantaneous estimate, then, simplify it as

$$
\partial w / \partial J(w_{l-1}) = \partial w / \partial \left[ E(i)^4 / \|u_l^4\|^4 \right]
$$

According to (25), we can obtain the following equation

$$
E(i) * \widetilde{E}(i) * E(i) * \widetilde{E}(i)
= \sum_{s=0}^{2^n-1} \tilde{\lambda}_s E_s * \sum_{s=0}^{2^n-1} \lambda_s E_s
= \sum_{s=0}^{2^n-1} \lambda_s \tilde{\lambda}_s
= \sum_{s=0}^{2^n-1} E_s^2 \lambda_s \tilde{\lambda}_s
$$

Therefore, plugging (33) and (41) into (40) results in

$$
\partial_w J(w_{l-1}) = \partial_w \left[ \frac{|E(i)|^4}{\|u_l^4\|^4} \right]
= \sum_{l=0}^{2^n-1} \lambda_l \partial_w l E_s^4
= \left\{ \frac{\|u_l^4\|^4}{\|u_l^4\|^4} \right\}
$$

we know $\partial_w l E_s^4$ can be obtained by using the estimation error in (29) as follows:

$$
\partial_w l E_s^4 = 4E_s^3 \partial_w l E_s = 4E_s^3 \partial_w l (D_s - \tilde{D}_s)
= -4E_s^3 \partial_w l \tilde{D}_s
$$

since $D_s$ does not depend on the weight array $w$, so the $\partial_w l D_s = 0$, then plugging (34) into (43), we can obtain

$$
\partial_w l E_s^4 = 4E_s^3 \partial_w l D_s
= -4E_s^3 \sum_{l=0}^{2^n-1} \tilde{\lambda}_s (u_q^l w_l)
= -4E_s^3 \sum_{l=0}^{2^n-1} \tilde{\lambda}_s (u_q^l w_l)
= -4E_s^3 \sum_{l=0}^{2^n-1} \tilde{\lambda}_s (u_q^l w_l)
$$

Noticing that $\partial w / \partial J(w_{l-1})$ will be different from zero only when $l = t$, thus, adopting the Kronecker delta function [38], $\partial w / \partial J(w_{l-1}) = \delta_{t,l}$. Finally, substituting (44) into (42), the stochastic gradient is obtained

$$
\partial_w J(w_{l-1}) = \frac{\sum_{l=0}^{2^n-1} \lambda_l \partial_w l E_s^4}{\|u_l^4\|^4}
$$

According to the GAAFs literature, we get the GA-NLMF update rule.

$$
w_l = w_{l-1} + \mu \left( E(i) * \widetilde{E}(i) * E(i) \right) /
\left\{ \|u_l^4\|^4 \right\}
$$

where $\mu$ is the step size of the adaptive filter algorithm.

IV. SIMULATION ANALYSIS

We know that signal prediction is one of the core functions of adaptive filtering, therefore, in this section, we tested the performance of the proposed algorithm on one step ahead prediction. The experiment mainly conducts several comparative experiments, they are as follows. Part A, we choose 3D Lorenz attractor to test the performance of NLMS, NLMF, GA-NLMS, GA-LMS, GA-LMK, and GA-NLMF algorithms. First, we compare NLMS [5] with GA-NLMS on the step ahead prediction problem. Second, we compare NLMF [12] with GA-NLMF on one step ahead prediction problem. Third, we also compare GA-NLMS and GA-LMS [20] on one step ahead prediction problem. The fourth, we compared GA-NLMF with GA-LMK algorithms. Finally, we analyze the performance of GA-NLMS and GA-NLMF on one step ahead prediction problem.
algorithms. Part C, we choose a complex value signal test the performance of GA-LMK and GA-NLMF algorithms.

As with linear algebra, the steady-state error of a GA must be a scalar value. Therefore, in the experiment, in order to be able to compare the convergence performance of different algorithms and the size of steady-state error, we use the magnitude of the mean absolute error to compare the difference between the estimated signal estimated by each algorithm and the actual signal.

A. 3D LORENZ ATTRACTOR

The Lorenz attractor is a 3-dimensions system used to model atmospheric turbulence, lasers and so on, it can be defined as follow by using following differential equations [41]:

\[
\begin{align*}
\frac{dx}{dt} &= \alpha (y - x) \\
\frac{dy}{dt} &= x(\rho - z) - y \\
\frac{dz}{dt} &= xy - \beta z
\end{align*}
\] (47)

we set the value of \( \alpha, \rho \) and \( \beta \) as 10, 28, 8/3.

1) NLMS AND GA-NLMS ALGORITHM

In this experiment, the NLMS and GA-NLMS algorithms are used to track the 3-dimensional Lorenz attractor as a comparative experiment. The step-size of NLMS and GA-NLMS algorithms are chosen to be \( \mu_{NLMS} = 2 \times 10^{-2} \), \( \mu_{GANLMS} = 10^{-2} \). Fig.1 shows the tracking performance of NLMS and GA-NLMS algorithms for different components of 3-dimensional Lorenz attractor. Fig.3 shows the signals of 3-dimensional Lorenz attractor are estimated by using NLMS and GA-NLMS algorithms respectively.

From Fig.1, we know that the GA-NLMS can track each component of the signal more quickly than NLMS algorithm. In Fig.3, we can see that the estimated signal is closer to the actual signal by GA-NLMS than NLMS algorithm. The mainly reason is that in the framework of geometric algebra, a multidimensional signal is represented as a multivector, and the algorithm based on GA can process multidimensional signal a holistic way, and remains the relationship among each component of the multidimensional signal, the estimated signal obtained by GA-NLMS is closer to the original signal. Thus, GA-NLMS works better than NLMS algorithm.

2) NLMF AND GA-NLMF ALGORITHM

In this experiment, the NLMF and GA-NLMF algorithms are used to track the 3-dimensional Lorenz attractor as a comparative experiment, the step-size of NLMF and GA-NLMF algorithms are chosen to be \( \mu_{NLMF} = 2 \times 10^{-2} \), \( \mu_{GANLMF} = 4 \times 10^{-3} \). Fig.2 shows the tracking performance of NLMF and GA-NLMF algorithms for each components of 3-dimensional Lorenz attractor. Fig.4 shows the signals of 3-dimensional Lorenz attractor are estimated by using NLMF and GA-NLMF algorithms respectively.

From Fig.4, we get the estimated signal of the lorentz attractor closer to the actual signal by using GA-NLMF algorithm, therefore, the GA-NLMF algorithm is more accurate than the NLMF algorithm. From Fig.2, we can see that GA-NLMF algorithm can track each component of the 3-dimensional signal more quickly than NLMF algorithm as the number of iterations increase.

It is clearly that GA-NLMF algorithm has a better predictive performance than NLMF algorithm. The main reason is that NLMF for multidimensional signal processing, it deals with each component of the multidimensional signal, it’s lost part of the multidimensional signal information, make the signal incomplete. GA-NLMF is different from NLMF algorithm, GA-NLMF represents a multidimensional signal as a multivector, which process multidimensional signal in a holistic way, and remains the relationship among each component of the multidimensional signal, therefore, GA-NLMS works better than NLMF algorithm.
3) GA-NLMS AND GA-LMS ALGORITHM
In this experiment, the GA-LMS and GA-NLMS algorithms are used to track the 3-dimensional Lorenz attractor, the step-size of GA-LMS and GA-NLMS algorithms are chosen to be $\mu_{\text{GALMS}} = 0.8 \times 10^{-5}$, $\mu_{\text{GANLMS}} = 4 \times 10^{-2}$. Fig.5 shows that the tracking performance of the GA-LMS and GA-NLMS algorithm for each component of the Lorenz attractor. Fig.8 shows that the estimated signal of the corresponding signal obtained by the GA-LMS and GA-NLMS algorithms.

From Fig.5 and Fig.8, we can know that GA-NLMS requires less iterations to track the actual signal than GA-LMS, and GA-NLMS tracks more accurately than GA-LMS algorithms. From [5], we know that the main drawback of LMS algorithm is that it is very sensitive to the input signal, and it is easy to be influenced by the input signal during the adaptive weight iteration, which makes the signal unstable. Although GA-LMS optimizes the LMS algorithm, enabling it to process multidimensional signals in a holistic way and ensuring the relationship among each components of multidimensional signals, its stability cannot be guaranteed with the increase of input signal. GA-NLMS algorithm is a variant of GA-LMS algorithm, which solves the problem by normalizing the input signal.

4) GA-NLMF AND GA-LMK ALGORITHM
In this experiment, the GA-LMK [22] and GA-NLMF algorithms are used to tracking the 3-dimensional Lorenz attractor, the step size of their algorithms are chosen to be $\mu_{\text{GANLMF}} = 4 \times 10^{-3}$, $\mu_{\text{GALMK}} = 10^{-6}$. Fig.6 shows that the tracking performance of the GA-LMK and GA-NLMF algorithms for each component of the Lorenz attractor. Fig.9 shows that the estimated signal of the corresponding signal obtained by the GA-LMK and GA-NLMF algorithms.
From the Fig.6, we can know that GA-NLMF algorithm can track each component of the 3-dimensional signal more quickly than GA-LMK as the number of iterations increase. The main reason is that the GA-NLMF algorithm is based on the normalized signal, which solves the sensitivity problem of the input signal and makes the algorithm more stable. In Fig.9, we can see that the estimated signal is closer to actual signal by GA-NLMF algorithm than GA-LMK algorithm. Besides, the learning curve of the GA-NLMF and GA-LMK algorithms is depicted in Fig.7. We can see that GA-NLMF has a faster convergence rate and a smaller mean absolute error than GA-LMK algorithm, that is to say, the performance of GA-NLMF algorithm is better than that of GA-LMK algorithm.

5) GA-NLMF AND GA-NLMS ALGORITHM
In this experiment, the GA-NLMF and GA-NLMS algorithms are used to track the 3-dimensional Lorenz attractor, the step-size of GA-NLMF and GA-NLMS algorithms are chosen to be $\mu_{GANLMF} = 4 \times 10^{-3}$, $\mu_{GANLMS} = 4 \times 10^{-2}$. Fig.10 shows that the tracking performance of the GA-NLMF and GA-NLMS algorithm for each component of the Lorenz attractor. Fig.11 shows that the estimated signal of the corresponding signals obtained by the GA-NLMF and GA-NLMS algorithms.

From Fig.10 and Fig.11, we can know that GA-NLMF requires less iterations to track the actual signal than GA-NLMS, and GA-NLMF tracks more accurately than GA-NLMS algorithms. From [42], [43], we know that NLMF has a faster convergence than NLMS algorithm, and NLMF is a higher-order statistic, making the weight update coefficient more robust, therefore the performance of NLMF is better than NLMS algorithm. Besides, the GA theory extends the NLMS and NLMF algorithm in GA space, allows them to process multidimensional signals in a holistic way, and remains the relationship among each component of the multidimensional signal. Compared with GA-NLMS, each component of the estimated signal obtained by GA-NLMF algorithm is closer to the actual signal, when the number of iterations is small. Therefore, the GA-NLMF has a faster convergence rate than GA-NLMS algorithm, the GA-NLMF works better than GA-NLMS in tracking performance.

The learning curve of the GA-NLMF, GA-NLMS and GA-LMS algorithms for Lorenz attractor is also depicted in Fig.7, we can see that GA-NLMF has a faster convergence rate and a smaller mean absolute error than GA-NLMS algorithm when the number of iteration is small. In the case of small step size, the convergence rate of GA-NLMF is significantly faster than that of GA-NLMS, that is to say, the GA-NLMF algorithm solves the trade-off between steady state error and convergence rate in the adaptive filtering domain. In Fig.7, we can also see that the mean absolute error obtained by GA-LMS algorithm fluctuates relatively large and unstable, the mainly reason is that it doesn’t
normalize the input signal and the weight update is affected by it. In addition, the GA-NLMF algorithm’s cost function is based on the four power of the normalized error signal which contains higher-order statistics. Therefore, GA-NLMF also has a faster convergence rate and smaller mean absolute error than GA-LMS algorithm. Besides, compared with
FIGURE 11. Tracking performance of GA-NLMF and GA-NLMS algorithm on the prediction of Lorenz attractor.

FIGURE 12. Tracking performance of GA-LMK and GA-NLMF algorithm on the prediction of each component of 3-dimensions Multiscroll attractor.

GA-LMS and GA-NLMS algorithms, the performance of GA-LMK algorithm is improved because its cost function is based on the negative kurtosis of error signals containing higher-order statistics. Experiments show that the performance of GA-NLMF algorithm is better than that of GA-NLMS, GA-LMK and GA-LMS algorithms.

B. 3D MULTISCROLL ATTRACTOR

The Multiscroll attractors are widely used in secure communications [44], it is defined by using the following differential equations [45].

\[
\begin{align*}
\frac{dx}{dt} &= \alpha(y - x) \\
\frac{dy}{dt} &= (c - \alpha)x - xz + cy \\
\frac{dz}{dt} &= xy - bz
\end{align*}
\]  

(48)

we set the value of the \(\alpha, b\) and \(c\) as 40, 3, and 28.

In this section, GA-LMS, GA-NLMS, GA-LMK, GA-NLMF algorithms are used to track the 3D Multiscroll attractor, the step-size of GA-NLMF and GA-LMK algorithms are chosen to be \(\mu_{GANLMF} = 4 \times 10^{-3}, \mu_{GALMK} = 10^{-6}\). Fig.12 shows that the tracking performance of the GA-LMK and GA-NLMF algorithms for each component of the Multiscroll attractor.

Fig.13 shows that the estimated signal of the corresponding signals obtained by the GA-LMK and GA-NLMF algorithms. The learning curve of the GA-LMS, GA-NLMS, GA-LMK and GA-NLMF algorithms for Multiscroll attractor is depicted in Fig.14.

From Fig.12 and Fig.13, we can know that GA-NLMF requires less iterations to track the actual signal than GA-LMK, and GA-NLMF tracks more accurately than GA-LMK algorithms. In Fig.14, we can see that GA-NLMF has a faster convergence rate and a smaller mean absolute error than the other algorithms. The significant performance gain of GA-LMK algorithm over GA-LMS and GA-NLMS algorithms is because of its cost function based on the negated kurtosis of the error signal which includes high-order statistics. On the other hand, since the loss function of GA-NLMF also includes higher-order statistics, the computation is on the same order of magnitude, and the signal is normalized, it solves the sensitivity problem of input signal and makes the algorithm more stable. Although GA-LMK can process multidimensional signals in a holistic way, its stability cannot be guaranteed with the increase of input signals, and the weight update is affected by input signals. Therefore, we can see that the proposed GA-NLMF algorithm has a faster convergence rate as well as a smaller mean absolute that of GA-LMK. That is to say, the GA-NLMF outperforms the GA-LMK algorithm.

C. COMPLEX SIGNAL

In this experiment, we choose a complex value signal to track the performance of the GA-LMK and GA-NLMF
algorithms. The complex value signal is defined as $y = e^{t+20j}$. From [32], we know that the multivector structure of geometric algebra can be regarded as a generalization of complex numbers, thus, a complex value signal can represent a GA multivector.

Fig. 15 shows that the estimated signal of the corresponding signals obtained by the GA-LMK and GA-NLMF algorithms. From Fig. 15, in the case of a small number of iterations, we can know that the estimated signal is closer to the original signal by using GA-NLMF than GA-LMK algorithm. GA-NLMF has a faster convergence rate than GA-LMK. Thus, we can conclude that the tracking performance of GA-NLMF is better than GA-LMK algorithms.

D. COMPUTATIONAL COMPLEXITY

In this section, we compared the computational of the proposed algorithms in this paper and the existing algorithms, the approximate time is calculated by using Matlab on PC with Intel(R) Core(TM) i5-4210U CPU@1.7GHz. Table 1 shows that the elapsed time of GA-LMS, GA-NLMS, GA-LMK, and GA-NLMF algorithms. From Table 1, the high elapsed time for the GA-LMK and GA-NLMF is slightly longer than the elapsed time for GA-LMS algorithm, the main reason is that the loss function of GA-LMK and GA-NLMF is more complex and contains high-order statistics. We can conclude GA-LMK and GA-NLMF algorithms can improve the steady-state performance of the adaptive filter with some extra computational cost. Besides, the elapsed time of GA-NLMS is longer to that of GA-LMS algorithm, the mainly reason is that GA-NLMS increases the elapsed time by normalizing the input signal for each iteration. The elapsed time of GA-LMK and GA-NLMF is similar, because both of them contain high-order statistics, and the computation is on the same order of magnitude. In addition, since the GA-NLMF algorithm normalizes the input signal, the stability of the algorithm is guaranteed, that is to say, the stability of GA-NLMF is better than that of GA-LMK. Besides, we can also see from Fig. 14 has a faster convergence rate and mean absolute error than GA-LMK.
Thus, we can conclude that the performance of GA-NLMF is better than that of GA-LMK.

V. CONCLUSION

In this paper, two normalized adaptive filtering algorithms based on geometric algebra are proposed for multidimensional signal processing. GA-NLMS and GA-NLMF extend the original NLMS and NLMF algorithm from real domain to GA space. In GA space, a multidimensional signal can be represented as a multivector, which guarantees the possible relationship among each component of the multidimensional signal, however, the original normalization algorithm ignores the relationship between the components of multidimensional signals and loses part of the information, resulting in incomplete signals. Therefore, the performance of the GA-NLMS and GA-NLMF is better than the original NLMS and NLMF algorithms, the experiments also show that the algorithm based on GA is better than the original normalized adaptive filtering algorithm in steady-state error and convergence rate performance on one step ahead prediction problem. On the other hand, the proposed GA-NLMS algorithm is also better than GA-LMS in convergence rate performance, we know that compared with LMS algorithm, NLMS ensures the stability of the algorithm by normalizing the input signal. When the algorithm is extended to GA space, GA-NLMS performs better than GA-LMS algorithm, and the estimated signal obtained by GA-NLMS is more stable. Because NLMF is a high-order moments algorithm, it performs better than NLMS algorithm, therefore, the proposed GA-NLMF algorithm has faster convergence rate and lower and more stable steady-state error than GA-NLMS algorithms, GA-NLMF also better than GA-LMS algorithm. Besides, The error signal of GA-LMK algorithm also includes high-order statistics, which greatly improves the performance. However, compared with GA-NLMF algorithm, it does not normalize the input signal, and the weight update will be affected by the input signal. The stability of the algorithm is not as good as GA-NLMF algorithm. In the experiment, we also prove that GA-NLMF algorithm is better than GA-LMK algorithm.

Currently, we extend the original NLMF and NLMS algorithm in GA space, however, many other algorithms have not been extended to GA space. In order to better deal with multidimensional signals, we also hope to extend more adaptive filtering algorithms with good performance to GA space in the next work, and combine these algorithms with practical applications. On the other hand, we will continue to study the optimal adaptive step size and time-varying step size normalization algorithm based on geometric algebra, it is expected that the normalization algorithm based on geometric algebra can get more applications and become an efficient algorithm.

REFERENCES

[1] A. H. Sayed, Adaptive Filters. Hoboken, NJ, USA: Wiley, 2008.
[2] P. M. Clarkson, Optimal and Adaptive Signal Processing. Boca Raton, FL, USA: CRC Press, 1993.
[3] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias, Advanced Digital Signal Processing. New York, NY, USA: Macmillan, 1992.
[4] S. Haykin, Adaptive Filter Theory. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.
[5] J. Dhiman, S. Ahmad, and K. Gulia, “Comparison between adaptive filter algorithms (LMS NLM NLS),” Int. J. Sci. Eng. Technol. Res., vol. 2, no. 5, pp. 1100–1103, 2013.
[6] C. Paleologu, J. Benesty, S. L. Grant, and C. Osterwise, “Variable step-size NLMS algorithms designed for echo cancellation,” in Proc. 43rd Asilomar Conf. Signals, Syst. Comput., 2009, pp. 633–637.
[7] Y.-S. Choi, H.-C. Shin, and W.-J. Song, “Robust regularization for normalized LMS algorithms,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 627–631, Aug. 2006.
[8] D. P. Mandic, “A generalized normalized gradient descent algorithm,” IEEE Signal Process. Lett., vol. 11, no. 2, pp. 115–118, Feb. 2004.
[9] H.-C. Shin, A. H. Sayed, and W.-J. Song, “Variable step-size NLMS and affine projection algorithms,” IEEE Signal Process. Lett., vol. 11, no. 2, pp. 132–135, Feb. 2004.
[10] J. Benesty, H. Rey, L. R. Vega, and S. Tressens, “A nonparametric VSS NLMS algorithm,” IEEE Signal Process. Lett., vol. 13, no. 10, pp. 581–584, Oct. 2006.
[11] M. K. Chan and C. F. N. Cowan, “Using a normalised LMF algorithm for channel equalisation with co-channel interference,” in Proc. EUSIPCO, Sep. 2002, pp. 48–51.

[12] E. Eweda and A. Zerguine, “A normalized least mean fourth algorithm with improved stability,” in Proc. 44th Asilomar Conf. Signals Syst. Comput., Nov. 2010, pp. 1002–1005.

[13] A. Zerguine, M. K. Chan, T. Y. Al-Naffouri, M. Moinuddin, and C. F. N. Cowan, “Convergence and tracking analysis of a variable normalised LMF (XE-NLMF) algorithm,” Signal Process., vol. 89, no. 5, pp. 778–790, May 2009.

[14] M. M. U. Faiz and A. Zerguine, “Convergence and tracking analysis of the ε–NSRLMF algorithm,” in Proc. 38th IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), pp. 5657–5660, May 2013.

[15] Y. Li, W. Liu, X. Li, Q. Huang, and X. Li, “GA-SIFT: A new scale invariant feature transform for multispectral image using geometric algebra,” Inf. Sci., vol. 281, pp. 559–572, Oct. 2014.

[16] R. Wang, M. Shen, and W. Cao, “Multivector sparse representation for multispectral images using geometric algebra,” IEEE Access, vol. 7, pp. 12755–12767, 2019.

[17] R. Wang, K. Wang, W. Cao, and X. Wang, “Geometric algebra in signal and image processing: A survey,” IEEE Access, vol. 7, pp. 156315–156325, 2019.

[18] R. Wang, W. Zhang, Y. Shi, X. Wang, and W. Cao, “GA-ORB: A new efficient feature extraction algorithm for multispectral images based on geometric algebra,” IEEE Access, vol. 7, pp. 71235–71244, 2019.

[19] R. Wang, Y. Shi, and W. Cao, “GA-SURF: A new speeded-up robust feature extraction algorithm for multispectral images based on geometric algebra,” Pattern Recognit. Lett., vol. 127, pp. 11–17, Nov. 2019.

[20] W. B. Lopes, A. Al-Nuaimi, and C. G. Lopes, “Geometric-algebra LMS adaptive filter and its application to rotation estimation,” IEEE Signal Process. Lett., vol. 23, no. 6, pp. 858–862, Jun. 2016.

[21] A. Al-Nuaimi, E. Steinbach, W. B. Lopes, and C. G. Lopes, “6DOF point cloud alignment using geometric algebra-based adaptive filtering,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2016, pp. 1–9.

[22] R. Yang, H. C. Huang, X. Wang, and W. Cao, “A novel least-mean-kurtosis adaptive filtering algorithm based on geometric algebra,” IEEE Access, vol. 7, pp. 78298–78310, 2019.

[23] R. Wang, M. Shen, T. Wang, and W. Cao, “L1-norm minimization for multi-dimensional signals based on geometric algebra,” Adv. Appl. Clifford Algebras, vol. 29, no. 2, p. 33, Apr. 2019.

[24] W. Rui, S. Miaomiao, W. Xiangyang, and C. WenMing, “RGA-CNNs: Convolutional neural networks based on reduced geometric algebra,” Sci. China Inf. Sci., to be published, doi: 10.1007/s11432-018-1513-5.

[25] R. Wang, Z. Cao, X. Wang, W. Xue, and W. Cao, “GA-STIP: Action recognition in multi-channel videos with geometric algebra based spatio-temporal interest points,” IEEE Access, vol. 6, pp. 56575–56586, 2018.

[26] M. Shen, R. Wang, and W. Cao, “Joint sparse representation model for multi-channel image based on reduced geometric algebra,” IEEE Access, vol. 6, pp. 24213–24223, 2018.

[27] M. J. Crowe, A History of Vector Analysis: The Evolution of the Idea OfA Vectorial System (Dover Books on Mathematics Series). New York, NY, USA: Dover, 1967.

[28] J. V. Jayme and R. D. Rocha, An Introduction to Clifford Algebras and Spinors. New York, NY, USA: Oxford Univ. Press, 2016.

[29] H. Grassmann and London Mathematical Society, Ausdehnungslehre, History of Mathematics. Providence, NJ, USA: AMS, 2000.

[30] D. Hestenes, “Clifford algebra to geometric calculus: A unified language for mathematics and physics,” Amer. J. Phys., vol. 33, no. 5, pp. 510–511, May 1985.

[31] C. H. Hua, T. Zeng, and A. J. M. Spencer, “A history of vector analysis,” J. Soc. Instrum. Control Eng., vol. 51, no. 4, pp. 338–350, Apr. 2012.

[32] W. B. Lopes and C. G. Lopes. (2016). Geometric-Algebra Adaptive Filters. [Online]. Available: http://www.teses.usp.br/teses/disponiveis/3/3142/tde-22092016-143525/en.php

[33] H. Grassmann and London Mathematical Society, Ausdehnungslehre, History of Mathematics. Providence, NJ, USA: AMS, 2000.

[34] M. Jiang, W. Liu, and Y. Li, “A general quaternion-valued gradient operator and its applications to computational fluid dynamics and adaptive beamforming,” in Proc. 19th Int. Conf. Digit. Signal Process., Aug. 2014, pp. 821–826.

[35] D. P. Mandic, Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models. Hoboken, NJ, USA: Wiley, 2009.

[36] C. C. Took and D. P. Mandic, “The quaternion LMS algorithm for adaptive filtering of hypercomplex processes,” IEEE Trans. Signal Process., vol. 57, no. 4, pp. 1316–1327, Apr. 2009.

[37] W. Wang, H. Zhao, and X. Zeng, “Geometric algebra correntropy: Definition and application to robust adaptive filtering,” IEEE Trans. Circuits Syst. II, Exp. Briefs, early access, Jul. 29, 2019, doi: 10.1109/TCSII.2019.2931507.

[38] J. A. K. Suykens and J. Vandewalle, “Generation of n-double scrolls (n = 1, 2, 3, 4, …),” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 40, no. 11, pp. 861–867, Nov. 1993.