Research on Robustness of Geometric Algebra Based Adaptive Filtering Algorithms In Non-Gaussian Environment

Rui Wang  
Shanghai University  
https://orcid.org/0000-0002-7974-9510

Yi Wang  
Shanghai University

Yanping Li  (yanpingli@shu.edu.cn)  
Shanghai University

Wenming Cao  
Shenzhen University

Research

Keywords: Adaptive filtering, Geometric algebra, MEE, MSE, α-stable noise

Posted Date: November 1st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1013240/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Research on robustness of geometric algebra based adaptive filtering algorithms in non-Gaussian environment

Rui Wang¹, Yi Wang¹, Yanping Li²* and Wenming Cao³

¹Correspondence: yanpingli@shu.edu.cn
²School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
Full list of author information is available at the end of the article

Abstract

In this paper, two new geometric algebra (GA) based adaptive filtering algorithms in non-Gaussian environment are proposed, which are deduced from the robust algorithms based on the minimum error entropy (MEE) criterion and the joint criterion of the MEE and the mean square error (MSE) with the help of GA theory. Some experiments validate the effectiveness and superiority of the GA-MEE and GA-MSEMEE algorithms in $\alpha$-stable noise environment. At the same time, the GA-MSEMEE algorithm has faster convergence speed compared with the GA-MEE.

Keywords: Adaptive filtering; Geometric algebra; MEE; MSE; $\alpha$-stable noise

1 Introduction

Adaptive filtering algorithms have a strong effect on signal processing, such as adaptive beamforming [1], acoustic echo cancelation [2–4], noise cancelation [5] and channel equalization [6]. Among them, mean square error (MSE) has been the typical criterion of adaptive filtering algorithms. Owing to its simple structure and rapid convergence, the LMS algorithm on the basis of MSE proposed by Widrow and Hoff has been applied in many fields [7–9]. Nevertheless, the performance of the LMS algorithm is not optimal, one problem is that the algorithm is vulnerable to the input signal, the other problem is the contradictory between step size and steady-state error. Subsequently, the NLMS algorithm put forward to solve these problems by normalizing the power of the input signal [10]. However, when signals are disturbed by abnormal values such as impulse noise, the performance of the LMS-type algorithm will be seriously degraded. Therefore, some robustness criteria have been proposed and successfully applied to adaptive filtering algorithms to deal with adaptive signal problems under impulsive noise, such as adaptive wireless channel tracking [11] and blind source decomposition [12]. Some typical robustness criteria include maximum correntropy criterion (MCC) [13–15], minimum error entropy (MEE) [16–18], generalized MCC [19] and minimum kernel risk-sensitive loss criterion [20]. They are insensitive to large outliers, which improves the performance under impulsive noise. In this regard, Professor J.C. principle and his team proposed to use the error signal of Renyi entropy instead of the MSE. Minimum error entropy is capable of getting better error distribution according to [21]. Although MEE criterion can obtain high accuracy, it does not take the mean factor into account, while the characteristics of MSE are just opposite to that of MEE. In this regard,
B. Chen et al. [21] put forward a joint criterion building up a connection between MSE and MEE by adding the weight.

In addition, recent studies have shown that MEE criterion is superior to MCC criterion and can be used for adaptive filtering [18] and Kalman filtering [17]. Therefore, G. Wang et al. [22] proposed recursive MEE, which has strong robustness under impulsive noise. However, the current adaptive filtering algorithms based on MEE criterion and the joint criterion of MSE and MEE only can be used for one-dimensional signals processing. It is worth noting that combined with geometric algebra, these algorithms can be extended to higher dimensions, so that the correlation of each dimension can be considered in the process of analyzing problems, and the performance of algorithms can be effectively improved.

Geometric algebra (GA) gives an effectiveness computing framework for multi-dimensional signal processing [23,24]. GA has a wide range of applications, such as image processing, multi-dimensional signal processing [25–28] and computer vision [29,30]. Combined with this framework, some scholars propose GA adaptive filtering algorithms. Lopes et al. [31] devised the GA-LMS algorithm and analyzed the feasibility of the algorithm, used for point cloud registration through rotation estimation. After that, Al-Nuaimi et al. [32] further exploited the potential of the algorithm, applied for the 6DOF alignment. Recently, W. Wang et al. [33] conducted the instantaneous performance analysis of GA-LMS. However, the LMS algorithm extended to the GA space still has some limitations, such as its poor performance under non-Gaussian noise. The $\alpha$-stable distribution fits very well with the actual data, and is consistent with multichannel interference in wireless networks and backscatter echoes in radar systems. Therefore, the use of $\alpha$-stable distribution to simulate non-Gaussian noise has more general significance. W. Wang et al. [34] deduced and proposed GA-MCC algorithm, analyzing its performance in $\alpha$-stable noise. The results show that GA-MCC has good robustness, but there is still room for improvement in its convergence rate. Due to the superiority of MEE criterion over MCC criterion, GA-MEE and GA-MSEMEE algorithms are proposed in this paper to improve the validity of existing GA adaptive filtering algorithms and expand the scope of application.

Our contributions are as follows. Firstly, according to the GA theory, the multi-dimensional problem is transformed into mathematical description, represented by multivectors. Secondly, the algorithms based on MEE criterion and its joint criterion with MSE is deduced in GA space. The original MEE and its joint criterion algorithm with MSE can be used for higher dimensional signals processing with the help of GA theory; Finally, some experiments validate the validity and the robustness of the GA-MEE and GA-MSEMEE algorithm.

The rest of this paper is arranged as follows. Section 2 introduces the related theories of geometric algebra and the traditional algorithms, which are obtained on the basis of MEE criterion and the joint criterion of MSE and MEE. Section 3 gives the derivation process of GA-MEE and GA-MSEMEE. The Experimental analysis of the two new algorithms in non-Gaussian environment compared with other GA based algorithms is provided in Section 4. Section 5 concludes this paper.
2 Related works

2.1 Basic theory

Geometric Algebra was proposed by the British mathematician William Kingdon Clifford, which provides a unified computational framework for multi-dimensional signal processing [35–37]. This algebraic approach contains all geometric operators and permits specification of constructions in a coordinate-free manner [38]. Compared with several particular cases of vector and matrix algebras, complex numbers and quaternions, using geometric algebra can deal with higher dimensional signals.

Assuming that an orthogonal basis of \( \mathbb{R}^n \) is \( \{e_1, e_2, \ldots, e_n\} \), the basis of \( G_n \) can be generated by multiplying the \( n \) basis elements (plus the scalar 1) via geometric product. The geometric product of two basis elements is non-commutative, its property is defined as:

\[
e_i e_j = e_{ij} = -e_{ji}, i, j = 1, \ldots, n, \forall i \neq j
\]  

\[
e_i e_j = e_i e_j = e_j, i, j = 1, \ldots, n, i \neq j
\]

Given \( n = p + q \), the expression of the operation rule of orthonormal basis is:

\[
e_i^2 = \begin{cases} 
1, & 1 \leq i \leq p \\
-1, & p + 1 \leq i \leq n
\end{cases}
\]

Thus, the basis of \( G_n \) is:

\[
\{1, e_i, e_i e_j, \cdots, e_1 e_2 \cdots e_n\}
\]

The core product in GA space is geometric product. The expression of the geometric product of vector \( a \) and \( b \) is:

\[
ab \triangleq a \cdot b + a \wedge b
\]

in which \( a \cdot b \) represents the inner product, which is commutative, \( a \wedge b \) denotes the outer product, which is not commutative. According to their properties, the following expression can be obtained:

\[
\begin{align*}
a \cdot b &= \frac{1}{2}(ab + ba) \\
a \wedge b &= \frac{1}{2}(ab - ba)
\end{align*}
\]

Suppose \( A \) is a general multivector in \( G_n \), the basic element of \( G_n \) can be defined as:

\[
A = \langle A \rangle_0 + \langle A \rangle_1 + \langle A \rangle_2 + \cdots = \sum_s \langle A \rangle_s
\]

which is made up of its \( s \)-vector part \( \langle \cdot \rangle_s \).
Actually, any multivector can be decomposed according to [39]:

\[
B = \sum_{s=0}^{2^n-1} e_s (e^s B) = \sum_{s=0}^{2^n-1} e_s \langle e_s B \rangle = \sum_{s=0}^{2^n-1} e_s B^s
\]  

(8)

In the operation of geometric algebra, the main properties used are as follows:

1. Scalar product: \( A^* B = \langle AB \rangle_0 \)
2. Cyclic reordering: \( AB \cdots C = \langle B \cdots CA \rangle \)
3. Clifford reverse: \( A \triangleq \sum_{s=0}^n (-1)^{s(s-1)/2} \langle A \rangle_s \)
4. Magnitude: \( |A| \triangleq \sqrt{A^* A} = \sqrt{\sum_s |\langle A \rangle_s|^2} \)

2.2 The related adaptive filtering algorithms

2.2.1 The MEE algorithm

Professor J.C. Principe and his research team proposed to replace MSE with the error signal of Renyi entropy in the training of supervised adaptive systems, this method uses a nonparametric estimator—Parzen window to estimate the probability density of a random variable directly from the sample points.

The Renyi entropy of the error sample is defined as:

\[
H_\alpha(e) = \frac{1}{1-\alpha} \log \int f_\alpha^\alpha(e) \, de
\]

\[
f_\alpha^\alpha(e) = \mathbb{E} \left[ f_{e_\alpha}^{\alpha^{-1}}(e) \right] \approx V_\alpha(e_k)
\]

\[
V_\alpha(e_k) = \left[ \frac{1}{T} \sum_{i=k-L}^{k-1} k_\alpha (e_k - e_i) \right]^{-1}
\]

(9)

Where \( \alpha > 0 \) is the order of entropy, \( V_\alpha(e_k) \) is information potential, and when \( \alpha \to 1 \), Renyi entropy is equivalent to Shannon entropy. In addition, to keep the orientation consistent with the LMS algorithm (minimization), select \( \alpha < 1 \). In this case, the minimum error entropy can be converted into minimizing the information potential.

Hence, for the traditional minimum error entropy (MEE) algorithm, its core expressions are:

\[
J(n) = V_\alpha(e_n)
\]

\[
w(n+1) = w(n) - \mu \left\{ 1 - \alpha \left[ \frac{1}{T} \sum_{i=n-L}^{n-1} k_\alpha (e(n) - e(i)) \right]^{\alpha-2} \right\}
\]

(10)

2.2.2 The MSE-MEE algorithm

The mean square error standard has good sensitivity. The minimum error entropy has a good error distribution, especially in the case of high-order statistics. Therefore, based on these two methods, a new performance index is proposed, which combines the advantages of each method to realize the sensitivity and the synchronization validity of the error distribution.

The core expressions of the LMS algorithm are:

\[
J(n) = \mathbb{E} \{ e^2(n) \}
\]

\[
w(n+1) = w(n) + 2\mu e(n)x(n)
\]

(11)
While the MSE-MEE algorithm is the mixed of square power of LMS and information potential of MEE. Then the MSE-MEE cost function is:

$$J(n) = \mathbb{E} \{ \eta e^2(n) + (1 - \eta)V_\alpha(e) \}$$

(12)

in which $\eta$ is the mixing parameter and $\eta \in [0,1]$.

Then the corresponding gradient algorithm is:

$$w(n + 1) = w(n) + \mu \cdot \left\{ 2\eta e(n)x(n) - (1 - \eta)(1 - \alpha) \left[ \frac{1}{L} \sum_{i=n-L}^{n-1} k_\sigma(e(n) - e(i)) \right]^{\alpha - 2} \right\}$$

(13)

### 3 Methods

#### 3.1 Problem formulation of adaptive filtering

Regarding the linear filtering model, its formulation involves the input signal of length $L$ $u(n)=[U_n, U_{n-1}, \cdots, U_{n-L+1}]^T$, the system vector to be estimated $w_o=[W_{o1}, W_{o2}, \cdots, W_{oL}]^T$, the weight vector $w(n)=[W_1(n), W_2(n), \cdots, W_L(n)]^T$ and the desired signal $d(n)$:

$$d(n) = u(n)^H w_o + v_n = \sum_{i=1}^{L} \hat{U}_{n-i+1} W_{oi} + v_n$$

(14)

In this research, we give some assumptions as follows:

A1) The multivector valued components of the input signal $u(n)$ are zero-mean white Gaussian processes with variance $\sigma^2_s$.

A2) The multivector valued components of the additive noise are described by $\alpha$-stable processes. $\alpha$-stable distribution is a family of four parameter distributions, which can be represented by S ($\alpha$, $\beta$, $\gamma$, $\sigma$), in which $\alpha$ denotes the characteristic index, which describes the tail of the distribution; $\beta$ denotes the skewness, $\gamma$ denotes the dispersion coefficient, $\sigma$ denotes the distribution position.

A3) The noise $v_n$, the initial weight vector $w_o$, the input signal $u(n)$ and the weight error vector $\Delta w_n$ are uncorrelated.

#### 3.2 The proposed GA-MEE algorithm

In this part, we deduce the GA-MEE algorithm with the help of GA theory [34]. In traditional algorithms, the cost function of MEE is expressed by information potential. When $\alpha \in (0,1)$, the minimum error entropy is equal to minimize the cost function. The GA-MEE cost function can be obtained by rewriting the formula (9) in the GA form.

$$J(w_{i-1}) = V_\alpha(e_i) = \left[ \frac{1}{L} \sum_{l=i-L}^{i-1} k_\sigma(E(i) - E(l)) \right]^{\alpha - 1}$$

(15)

in which $E(i) = D(i) - \hat{D}(i)$, $\hat{D}(i) = u_i^T w_{i-1}$, $L$ denotes the length of the sliding window, $k_\sigma(x)$ denotes the Gaussian kernel defined as $k_\sigma(x) = \exp \left( -\frac{x^2}{\gamma} \right)$, where $\gamma$ is the kernel size.
Our algorithms keep the same direction as the LMS algorithm, which is opposite to that of the steepest-descent rule [40], yielding the adaptive rule based on GA:

$$w_i = w_{i-1} - \mu B \partial_w J (w_{i-1})$$  \hspace{1cm} (16)$$

where B denotes a multivectors matrix. Choosing different B, we will get various types of adaptive filtering algorithm [40]. Let B be the identity matrix here.

The derivative term $\partial_w J (w_{i-1})$ in the (10) can be calculated as:

$$\partial_w J (w_{i-1}) = \left[ \frac{1}{L} \sum_{l=i-L}^{i-1} \exp \left( -\frac{|E(i) - E(l)|^2}{\sigma^2} \right) \right]^{\alpha - 1}$$

$$ = \left\{ \left[ \frac{1}{L} \sum_{l=i-L}^{i-1} \exp \left( -\frac{|E(i) - E(l)|^2}{\sigma^2} \right) \right]^{\alpha - 1} \right\}$$

(17)

where $|E(i) - E(l)|^2$ is given by

$$|E(i) - E(l)|^2 = (E(i) - E(l)) \ast (E(i) - E(l))$$

$$= E(i) \ast E(i) - E(i) \ast E(l) - E(l) \ast E(i) + E(l) \ast E(l).$$

(18)

According to the formula (8), the differential operator $\partial_w$ can be expressed in another form. Thus, we obtain the new expression of $\partial_w$:

$$\partial_w = \sum_{k=1}^{2^n} \gamma_k \langle \gamma_k \partial_w \rangle = \sum_{k=1}^{2^n} \gamma_k \partial_{w,k}$$

(19)

in which $\partial_{w,l}$ is the common derivative from standard calculus and only relates to blade l, $\{ \gamma_l \}$ is the basis of $\mathbb{G}_n$.

Similarly, given $\tilde{D}(i) = u^T_i w_{i-1}$, then $\tilde{D}(i)$ can be expanded as follows according to (8):

$$\tilde{D}(i) = u^*_i w_{i-1} = \sum_{A=1}^{2^n} \gamma_A \langle \gamma_A (u^*_i w_{i-1}) \rangle$$

(20)

Since $u_i$ and $w_{i-1}$ are arrays with M multivector entries, they can be decomposed as follows by employing (8),

$$u^*_i = \sum_{A=1}^{2^n} (u^*_i \gamma_A) \tilde{\gamma}_A = \sum_{A=1}^{2^n} u^T_{i,A} \tilde{\gamma}_A$$

(21)

and

$$w_{i-1} = \sum_{A=1}^{2^n} \gamma_A \langle \gamma_A w_{i-1} \rangle = \sum_{A=1}^{2^n} \gamma_A w_{i-1,A}$$

(22)
Plugging (22) and (23) back into (21),

\[
\hat{D}(i) = u_i^* w_{i-1} = \sum_{A=1}^{2^n} \gamma_A \langle \tilde{\gamma}_A (u_i^* w_{i-1}) \rangle
\]

\[
= \sum_{A=1}^{2^n} \gamma_A \left( \tilde{\gamma}_A \left( \sum_{B=1}^{2^n} u_B^T \gamma_B \sum_{C=1}^{2^n} \gamma_C w_C \right) \right)
\]

\[
= \sum_{A=1}^{2^n} \gamma_A \sum_{B,C=1}^{2^n} \langle \tilde{\gamma}_A (u_B^T \gamma_B \gamma_C w_C) \rangle \tag{23}
\]

\[
= \sum_{A=1}^{2^n} \gamma_A \sum_{B,C=1}^{2^n} \langle \tilde{\gamma}_A \gamma_B \gamma_C \rangle (u_B^T \cdot w_C)
\]

\[
= \sum_{A=1}^{2^n} \gamma_A \hat{D}_A \tag{24}
\]

in which

\[
\hat{D}_A = u_i^* w_{i-1} = \sum_{B,C=1}^{2^n} \langle \tilde{\gamma}_A \gamma_B \gamma_C \rangle (u_B^T \cdot w_C), A = 1, \ldots, 2^n
\]

Thus, the derivative term \( \partial_w |E(i) - E(l)|^2 \) in (18) can be calculated as:

\[
\partial_w |E(i) - E(l)|^2 = \partial_w |E(i)|^2 + \partial_w |E(l)|^2 - 2 \partial_w \langle E(i) \tilde{E}(l) \rangle \tag{25}
\]

According to (20), each term of equation (26) can be expanded:

\[
\begin{align*}
\partial_w |E(i)|^2 &= \left( \sum_{D=1}^{2^n} \gamma_D \partial_w \partial_D \right) \left( \sum_{A=1}^{2^n} \varepsilon^2_i \right) = \left( \sum_{A,D=1}^{2^n} \gamma_D \partial_w \partial_D \varepsilon^2_i \right) \\
\partial_w |E(l)|^2 &= \left( \sum_{D=1}^{2^n} \gamma_D \partial_w \partial_D \right) \left( \sum_{A=1}^{2^n} \varepsilon^2_i \right) = \left( \sum_{A,D=1}^{2^n} \gamma_D \partial_w \partial_D \varepsilon^2_i \right) \\
\partial_w \langle E(i) \tilde{E}(l) \rangle &= \left( \sum_{D=1}^{2^n} \gamma_D \partial_w \partial_D \right) \left( \sum_{A=1}^{2^n} \gamma_A \varepsilon_i \right) = \left( \sum_{A,D=1}^{2^n} \gamma_D \partial_w \partial_D \varepsilon_i \right) \\
&= \sum_{A,D=1}^{2^n} \gamma_D \partial_w \partial_D \varepsilon_i \varepsilon_i \tag{26}
\end{align*}
\]

in which

\[
\begin{align*}
\partial_w \partial_D \varepsilon^2_i &= 2 \varepsilon_i \left( \partial_w \partial_D \left( \varepsilon_i \varepsilon_i - \nabla_i \right) \right) = -2 \varepsilon_i \left( \partial_w \partial_D \nabla_i \right) \\
\partial_w \partial_D \varepsilon_i &= 2 \varepsilon_i \left( \partial_w \partial_D \left( \varepsilon_i \nabla_i - \nabla_i \right) \right) = -2 \varepsilon_i \left( \partial_w \partial_D \varepsilon_i \right) \\
\partial_w \partial_D \varepsilon_i \varepsilon_i &= \varepsilon_i \left( \partial_w \partial_D \varepsilon_i \right) + (\partial_w \partial_D \varepsilon_i \varepsilon_i) \varepsilon_i \varepsilon_i \\
&= -\varepsilon_i \left( \partial_w \partial_D \varepsilon_i \right) - \left( \partial_w \partial_D \varepsilon_i \varepsilon_i \right) \varepsilon_i \varepsilon_i \tag{27}
\end{align*}
\]

According to (25), the parts \( \nabla_i \) and \( \nabla_i \) in formula (28) can be expressed as:

\[
\begin{align*}
\nabla_i &= \sum_{B,C=1}^{2^n} \langle \tilde{\gamma}_A \gamma_B \gamma_C \rangle (u_B^T \cdot w_C), A = 1, \ldots, 2^n \\
\nabla_i &= \sum_{B,C=1}^{2^n} \langle \tilde{\gamma}_A \gamma_B \gamma_C \rangle (u_B^T \cdot w_C), A = 1, \ldots, 2^n \tag{28}
\end{align*}
\]
From (28), $\partial_w, D \hat{d}_{i,A}$ and $\partial_w, D \hat{d}_{i,A}$ need to be calculated, according to (26) and (29), $\partial_w, D \hat{d}_{i,A}$ can be unfolded as:

$$\partial_w, D \hat{d}_{i,A} = \partial_w, D \left[ \sum_{B,C=1}^{2^n} \langle \hat{\gamma}_A \hat{\gamma}_B \gamma_C \rangle (u_{iB}^T \cdot w_C) \right]$$

$$= \sum_{B,C=1}^{2^n} \langle \hat{\gamma}_A \hat{\gamma}_B \gamma_C \rangle \partial_w, D (u_{iB}^T \cdot w_C)$$

(29)

$$= \sum_{B,C=1}^{2^n} \langle \hat{\gamma}_A \hat{\gamma}_B \gamma_C \rangle \delta_{CD} u_{iB}^T$$

in the same way,

$$\partial_w, D \hat{d}_{i,A} = \sum_{B,C=1}^{2^n} \langle \hat{\gamma}_A \hat{\gamma}_B \gamma_C \rangle \delta_{CD} u_{iB}^T$$

(30)

Plugging (28), (30) and (31) into (27) result in

$$\partial_w |E(i)|^2 = -2e_{i,A} \left( \partial_w, D \hat{d}_{i,A} \right)$$

$$= -2 \sum_{A,D=1}^{2^n} \gamma_D e_{i,A} \sum_{B,C=1}^{2^n} \langle \hat{\gamma}_A \hat{\gamma}_B \gamma_C \rangle \delta_{CD} u_{iB}^T$$

$$= -2 \sum_{A,D=1}^{2^n} e_{i,A} \sum_{B,C=1}^{2^n} \gamma_D \langle \hat{\gamma}_A \hat{\gamma}_B \gamma_D \rangle u_{iB}^T$$

$$= -2 \sum_{A,D=1}^{2^n} e_{i,A} \gamma_D \langle \hat{\gamma}_A u_{i}^* \gamma_D \rangle$$

(31)

in the same way,

$$\partial_w |E(l)|^2 = -2u_l E(l)$$

(32)

and

$$\partial_w (E(i) \hat{E}(l)) = - \sum_{A,D=1}^{2^n} \gamma_D \left\{ \sum_{B,C=1}^{2^n} \langle \hat{\gamma}_A \hat{\gamma}_B \gamma_C \rangle \delta_{CD} u_{iB}^T \right\}$$

$$- \sum_{A=1}^{2^n} e_{i,A} \langle \hat{\gamma}_D u_{i}^* \gamma_A \rangle - \sum_{A=1}^{2^n} \gamma_D \langle \hat{\gamma}_D u_{i}^* \gamma_A \rangle e_{i,A}$$

(33)
Plugging (32), (33) and (34) into (26) yields the following expression:

$$\partial_w |E(i) - E(l)|^2 = -2u_i E(i) - 2u_i E(l) - 2 (-u_i E(i) - u_i E(l))$$

$$= -2u_i E(i) - 2u_i E(l) + 2u_i E(i) + 2u_i E(l)$$

$$= -2(u_i - u_l) (E(i) - E(l))$$  \( (34) \)

Finally, plugging (35) into (18), the gradient expression can be written as:

$$\partial_w J (w_{i-1}) = \left[ \frac{1}{L} \sum_{l=i-L}^{i-1} \exp \left( -\frac{|E(i) - E(l)|^2}{\sigma^2} \right) \right]^{\alpha-1}$$

$$= \left\{ \frac{2(\alpha-1)}{L^{\alpha} \sigma^2} \left[ \sum_{l=i-L}^{i-1} \exp \left( -\frac{|E(i) - E(l)|^2}{\sigma^2} \right) \right]^{\alpha-1} \right\}$$

Then plugging (36) into (17), we can obtain the GA-MEE updating rule:

$$w_i = w_{i-1} + \mu \left\{ \frac{2(\alpha-1)}{L^{\alpha} \sigma^2} \left[ \sum_{l=i-L}^{i-1} \exp \left( -\frac{|E(i) - E(l)|^2}{\sigma^2} \right) \right]^{\alpha-1} \right\}$$

in which \( \mu \) denotes the step size.

### 3.3 The proposed GA-MSEMEE algorithm

In the same way, according to GA theory, we can obtain the GA-MSEMEE cost function as follows by rewriting the formula (11) in the GA form.

$$J (w_{i-1}) = \mathbb{E} \left\{ \eta |E(i)|^2 + (1 - \eta)V_{\alpha} (e_i) \right\}$$  \( (37) \)

where \( \eta \) is the mixing parameter and \( \eta \in [0, 1] \).

When we replace the mathematical expectation of the preceding and subsequent terms of equation (38) with instantaneous value and sample average respectively, \( \partial_w J (w_{i-1}) \) can be expressed as:

$$\partial_w J (w_{i-1}) = \partial_w \left( \mathbb{E} \left\{ \eta |E(i)|^2 + (1 - \eta)V_{\alpha} (e_i) \right\} \right)$$

$$= \partial_w \left( \eta |E(i)|^2 + (1 - \eta)\partial_w V_{\alpha} (e_i) \right)$$

$$= \eta \partial_w |E(i)|^2 + (1 - \eta)\partial_w V_{\alpha} (e_i)$$  \( (38) \)

The former term of formula (39) is equivalent to GA-LMS algorithm, and the latter term of formula (39) is to seek deviation guide to information potential. In order to keep the whole direction consistent (minimized), select \( \alpha \in (0, 1) \). According to (32), \( \partial_w |E(i)|^2 \) is:

$$\partial_w |E(i)|^2 = -2u_i E(i)$$  \( (39) \)
According to (16) and (36), $\partial_w V_\alpha (e_i)$ is:

$$\partial_w V_\alpha (e_i) = \left\{ \frac{2(\alpha - 1)}{\sum_{i=1-L}^{i-1} \exp \left( - \frac{|E(i) - E(l)|^2}{\sigma^2} \right)} \cdot \left( u_i - u_l \right) (E(i) - E(l)) \right\}^{\alpha - 1} .$$

(40)

Plugging (40) and (41) into (39), we can obtain the GA-MSEMEE updating rule:

$$w_i = w_{i-1} + \mu \left\{ \frac{2\eta u_i E(i) + \frac{2(1-\eta)(1-\alpha)}{L^\sigma \sigma^2} \left[ \sum_{i=1-L}^{i-1} \exp \left( - \frac{|E(i) - E(l)|^2}{\sigma^2} \right) \right]^{\alpha - 1} \cdot \left( u_i - u_l \right) (E(i) - E(l)) \right\} .$$

(41)

in which $\mu$ denotes the step size, $\eta$ denotes the mixing parameter and $\eta \in [0, 1]$.

4 Results and discussion

This section carries out some experiments, analyzing the performance of the two new algorithms in non-Gaussian environment. First of all, in order to know how to select appropriate adjustable parameters for GA-MEE and GA-MSEMEE algorithms, the experimental part analyzes the influence of these parameters (the kernel width $\sigma$, the order of entropy $\alpha$, and weight coefficient $\eta$) on the MSD learning curves in detail. Secondly, GA-MEE and GA-MSEMEE algorithms are compared with other GA based algorithms to verify their superiority. Finally, the algorithms are applied to multi-dimensional signal denoising in non-Gaussian environment to verify their validity in processing multi-dimensional signals.

All MSD learning curves and the experimental data are averaged 50 independent runs. In this paper, initial weight vector $\omega_0$ denotes a $5 \times 1$ multivector, and the length of the sliding window is $L = 8$. The input signal and noise are shown in A1 and A2, $\alpha$-stable noise distribution is given by $S(1.5, 0, 1, 0)$ in the experiment. In addition, we use the generalized signal-to-noise ratio ($GSNR = 10\log_2 \left( \sigma^2 / \gamma_v \right)$) to describe the relationship between the input signal and noise, $\sigma^2$ is the variance of input signal multivector, $\gamma_v$ is the dispersion coefficient of noise.

4.1 The performance of GA-MEE and GA-MSEMEE algorithms under different parameters

Herein, we discuss the effect of the parameters $\sigma$, $\eta$ and $\alpha$ on the performance of the two new algorithms for 4-dimension signal. The performance of the two new algorithms are estimated by the mean-square deviation (MSD), $MSD = \mathbb{E} \left\{ \| w_0 - w(n) \|_2^2 \right\}$. According to the equation (37) and (42), the GA-MEE algorithm mainly involves the parameters $\sigma$ and $\alpha$, and the GA-MSEMEE algorithm mainly involves the parameters $\sigma$, $\eta$ and $\alpha$. In the following experiments, we select $\mu_{GA-MEE} = \mu_{GA-MSEMEE} = 0.5$ and $GSNR = 0dB$ for the GA-MEE and GA-MSEMEE.

4.1.1 GA-MEE algorithm

This section selects different parameters $\sigma$ and $\alpha$, then calculates the MSD of the GA-MEE algorithm under different parameters. Table 1 displays the steady-state MSDs under different parameters ($\sigma$ and $\alpha$) of the GA-MEE.
Table 1 The steady-state MSD of the GA-MEE algorithm

| α = 0.3 | α = 0.5 | α = 0.6 | α = 0.7 | α = 0.8 |
|---------|---------|---------|---------|---------|
| σ = 50  | -13.91  | -15.92  | -17.13  | -18.47  | -20.25  |
| σ = 60  | -15.84  | -17.47  | -18.38  | -19.52  | -21.06  |
| σ = 70  | -17.25  | -18.57  | -19.37  | -20.85  | -22.80  |
| σ = 90  | -18.46  | -20.04  | -21.96  | -22.50  | -23.50  |
| σ = 100 | -19.34  | -20.42  | -21.38  | -22.70  | -24.38  |

To further instinctively analyze the effect of kernel width and order of entropy on the GA-MEE algorithm, the steady-state MSD is taken as a function of kernel width and order of entropy are plotted in Fig.1 for various values of the kernel width σ and the order of entropy α.

The tendency of steady-state values in respect of kernel width and order of entropy are clearly highlighted in Fig.1. It can be obtained from Table 1 and the 3-dimensional diagram that the steady-state MSD is smaller with both larger values of σ and α.

Fig.2 demonstrates the instantaneous MSDs of the GA-MEE under various parameters. The GA-MEE1, GA-MEE2, GA-MEE3, GA-MEE4, and GA-MEE5 denote [α = 0.3, σ = 50], [α = 0.5, σ = 60], [α = 0.6, σ = 70], [α = 0.7, σ = 90] and [α = 0.8, σ = 100], respectively. Since increasing two parameters at the same time leads to the decrease of steady-state value and slow convergence rate, it is difficult to determine the role of a single parameter in the performance of the GA-MEE. Therefore, it is necessary to use the method of controlling variables.

Different parameter σ: The value of parameter α is setting as 0.6, and the value of parameter σ is setting as 50, 60, 70, 90, 100, respectively. Fig.3 shows the instantaneous MSDs of the GA-MEE under various σ. It can be seen from Fig.3, as kernel width increases, the steady-state MSD decreases and convergence rate increases. But when the parameter σ exceeds a certain value, the convergence rate decreases gradually. So, the selection of σ should balance the steady-state MSD and convergence rate. In this group of experiments, its convergence rate is the best when σ = 70.

Different parameter α: The value of parameter σ is setting as 70, and the value of parameter α is setting as 0.3, 0.5, 0.6, 0.7, 0.8, respectively. Fig.4 demonstrates the instantaneous MSDs of the GA-MEE under various α. The steady-state MSD increases with the increase of the order of entropy α, and the convergence rate decreases obviously. So, the selection of α should balance the steady-state MSD and convergence rate.

4.1.2 GA-MSEMEE algorithm

From the experimental part of the GA-MEE, it is concluded that the greater the parameter α, the slower the convergence rate. In order to study the influence of parameters on the GA-MSEMEE and the superiority of the GA-MSEMEE in convergence rate, this section selects different parameters σ and η, to analyze the performance of the GA-MSEMEE when α = 0.8. Table 2 displays the steady-state MSDs under different parameters (σ and η) of the GA-MSEMEE.
To further instinctively analyze the effect of kernel width and weight coefficient on the GA-MSEMEE algorithm, the steady-state MSD is taken as a function of kernel width and weight coefficient are plotted in Fig.5 for various values of kernel width $\sigma$ and weight coefficient $\eta$.

### Table 2 The steady-state MSD of the GA-MSEMEE algorithm

| $\sigma$ | $\eta = 0.3$ | $\eta = 0.5$ | $\eta = 0.6$ | $\eta = 0.7$ | $\eta = 0.8$ |
|---------|---------|---------|---------|---------|---------|
| 50      | -20.06  | -20.11  | -20.36  | -20.00  | -20.11  |
| 60      | -21.21  | -21.30  | -21.63  | -21.14  | -21.25  |
| 70      | -22.14  | -22.47  | -22.27  | -22.71  | -22.10  |
| 90      | -23.15  | -23.84  | -23.68  | -23.47  | -23.58  |
| 100     | -24.05  | -23.90  | -24.80  | -24.36  | -24.68  |

Fig.5 clearly shows the tendency of steady-state MSD in relation to kernel width and weight coefficient. It is shown as the Table 2 and 3-dimensional diagram that the steady-state value is smaller as $\sigma$ becomes larger. However, from the numerical point of view, the influence of the weight coefficient $\eta$ on MSD is not obvious.

Fig.6 shows the MSD learning curves of the GA-MSEMEE under various parameters, in which GA-MSEMEE1, GA-MSEMEE2, GA-MSEMEE3, GA-MSEMEE4, and GA-MSEMEE5 denote $[\eta = 9 \times 10^{-6}, \sigma = 50]$, $[\eta = 8.5 \times 10^{-6}, \sigma = 60]$, $[\eta = 8 \times 10^{-6}, \sigma = 70]$, $[\eta = 7.5 \times 10^{-6}, \sigma = 90]$ and $[\eta = 7 \times 10^{-6}, \sigma = 100]$, respectively. Since it is difficult to determine the role of a single parameter in the performance of the GA-MSEMEE, it is necessary to use the method of controlling variables.

Different parameter $\sigma$: The value of parameter $\eta$ is setting as $8.5 \times 10^{-6}$, and the value of parameter $\sigma$ is setting as 50, 60, 70, 90, 100, respectively. Fig.7 shows the instantaneous MSDs of the GA-MSEMEE under various $\sigma$. It is concluded from Fig.7 that as the kernel width becomes more larger, the steady-state MSD and convergence rate decrease gradually. Comprehensively considering the above two indicators, GA-MSEMEE has better performance when $\sigma = 70$ in this group of experiments.

Different parameter $\eta$: The value of parameter $\sigma$ is setting as 70. Since the values of parameters $\eta$ are similar in Table 2, it is difficult to see the impact of these parameters on the MSDs of the GA-MSEMEE. Thus, we set the parameters $\eta$ at large intervals, which are: $7 \times 10^{-6}$, $7 \times 10^{-5}$, $7 \times 10^{-4}$, $8 \times 10^{-4}$ and $9 \times 10^{-4}$. Fig.8 shows the instantaneous MSDs of the GA-MSEMEE under different $\eta$. As $\eta$ increases by ten times, the convergence rate becomes faster, the steady-state MSD gradually increases, and the robustness of the algorithm becomes worse. Therefore, the selection of weight coefficient should comprehensively compare the performance of three aspects. In this group of experiments, GA-MSEMEE has the best performance when $\eta = 7 \times 10^{-5}$.

### 4.2 Comparison of different GA based algorithms

In this part, we contrast the MSD learning curves of the two new algorithms to that of GA-LMS [31], GA-NLMS [41], GA-MCC [34] algorithms under different GSNR. Their parameters are set as follows: $\mu_{GA-LMS} = 8 \times 10^{-4}$, $\mu_{GA-NLMS} = 0.8$, $\mu_{GA-MMC} = 0.5(\sigma = 40)$, $\mu_{GA-MEE} = 0.5(\alpha = 0.1, \sigma = 90)$, $\mu_{GA-MSEMEE} =$
0.5(\alpha = 0.1, \sigma = 300, \eta = 0.0006), trying to make the convergence rate of each algorithm consistent. Fig. 9 demonstrates the instantaneous MSDs of different algorithms.

As can be seen from Fig. 9, compared with GA-MCC, the GA-MEE has better steady-state MSD and convergence rate, but its convergence rate slows down significantly with the decrease of GSNR. Compared with GA based LMS-type algorithms, the GA-MEE has better steady-state MSD and robustness, but GA-MEE needs more iterations to converge. The improved GA-MSEMEE algorithm solves this problem to a certain extent. The GA-MSEMEE always maintains superior convergence rate, good steady-state MSD and robustness under different GSNR.

4.3 Application and multi-dimensional signal analysis

In this part, the two new algorithms are applied to signal denoising. In order to test their superiority in non-Gaussian environment, we performed the following experiments.

Fig. 10 demonstrates the denoising results of 4-dimension signal with GA-LMS, GA-NLMS, GA-MCC, GA-MEE and GA-MSEMEE with GSNR=0dB. Their parameters are set as follows: \(\mu_{GA-LMS} = 7 \times 10^{-7}, \mu_{GA-NLMS} = 7 \times 10^{-4}, \mu_{GA-MCC} = 0.5(\sigma = 200), \mu_{GA-MEE} = 0.5(\alpha = 0.8, \sigma = 300), \mu_{GA-MSEMEE} = 0.5(\alpha = 0.1, \sigma = 300, \eta = 2 \times 10^{-6})\). As shown in Fig. 10, GA-LMS, GA-NLMS and GA-MCC algorithms all need an adaptive process at the beginning of denoising, which the proposed algorithms do not need. Fig.11 shows the average 4-dimensional signal recovery errors of different algorithms with different GSNR. The recovery error of 4-dimensional signal is described by \(\|u' - u\|_2^2\), which represents the norm square of the difference between the denoised signal and the clean signal.

What is more, it is worth noting that the two new algorithms can be applied to higher dimensional signal processing. Fig.12 demonstrates the denoising results of 8-dimensional signal with GA-MEE and GA-MSEMEE with GSNR = 0dB.

4.4 Computational Complexity

The running time of different algorithms for 4-dimensional and 8-dimensional signal denoising is shown in Table 3. The experiments are carried out via MATLAB with Intel (R) Core (TM) i7-6500U 2.50GHz CPU and 4 GB memory.

|          | GA-LMS | GA-NLMS | GA-MCC | GA-MEE | GA-MSEMEE |
|----------|--------|---------|--------|--------|-----------|
| 4-dimension | 0.7976 | 0.8526  | 1.5189 | 7.5411 | 7.8547    |
| 8-dimension | 1.5326 | 1.7636  | 3.1281 | 14.3391| 15.6693   |

Table 3 shows that the proposed algorithms in this paper have higher computational complexity. The reason for the higher computational complexity of GA-MEE algorithm is that it involves the calculation of minimum error entropy, which includes exponential operation of different error signals. The computational complexity of GA-MSEMEE is the highest, mainly because GA-MSEMEE algorithm is acquired by fusing MSE and MEE through weight coefficients.
5 Conclusions
Two new GA based algorithms GA-MEE and GA-MSEMEE are proposed, which are deduced from the MEE criterion and the joint criterion respectively, combined with GA theory. The GA-MEE and GA-MSEMEE show strong robustness and high precision for high-order signal processing in non-Gaussian environment. However, although the GA-MEE shows more robustness than other algorithms, its convergence rate and sensitivity are low. The GA-MSEMEE can effectively compensate for the lack of the GA-MEE. The experiments demonstrate that the GA-MSEMEE achieves a good balance between robustness and convergence rate.

Due to the high accuracy and sensitivity of the GA-MSEMEE, the algorithm can also be applied to more aspects, such as signal prediction, which can be further studied. Moreover, how to reduce the computational complexity is also a major direction of further research.

Abbreviations
GA: Geometric Algebra; MEE: Minimum Error Entropy; MSE: Mean Square Error; MSEMEE: a joint criterion of Mean Square Error and Minimum Error Entropy; LMS: Least Mean Square; NLMS: Normalized Least Mean Square; MCC: Maximum Correntropy Criterion; MSD: Mean-square deviation; GSNR: Generalized signal-to-noise ratio.

Declarations
Acknowledgements
Not applicable

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

Availability of data and materials
The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Authors’ contributions
RW proposed the new idea of the paper and participated in the outage performance analysis. YW performed the simulations and drafted the paper. YL and WC conceived of the study, and participated in its design and coordination and helped to draft the manuscript. All authors have read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Author details
1School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China. 2School of Communication and Information Engineering, Shanghai University, Shanghai, China. 3Office of Academic Affairs, Shanghai University, Shanghai, China. 4College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China.

References
1. Kuhn, E.V., Pitz, C.A., Matsuo, M.V., Bakri, K.J., Benesty, J.: A kronecker product clms algorithm for adaptive beamforming. Digital Signal Processing 111, 102968 (2021). doi:10.1016/j.dsp.2021.102968
2. Pauline, S.H., Samiappan, D., Kumar, R., Anand, A., Kar, A.: Variable tap-length non-parametric variable step-size nlms adaptive filtering algorithm for acoustic echo cancellation. Applied Acoustics 159, 107074 (2020). doi:10.1016/j.apacoust.2019.107074
3. Albu, F., Nakagawa, R., Nordholm, S.: Proportionate algorithms for two-microphone active feedback cancellation. In: 2015 23rd European Signal Processing Conference (EUSIPCO), pp. 290–294 (2015). doi:10.1109/EUSIPCO.2015.736391
4. Pogula, R., Kumar, T.K., Albu, F.: Robust sparse normalized lmat algorithms for adaptive system identification under impulsive noise environments. Circuits, Systems, and Signal Processing 38, 5103–5134 (2019). doi:10.1007/s00034-019-01111-3
5. Jayakumar, E.P., Sathidevi, P.S.: An integrated acoustic echo and noise cancellation system using cross-band adaptive filters and wavelet thresholding of multiband spectrum. Applied Acoustics 141, 9–18 (2018). doi:10.1016/j.apacoust.2015.05.029
6. Arablouei, R., Degangay, K.: Low-complexity adaptive decision-feedback equalization of mimo channels. Signal Processing 92(6), 1515–1524 (2012). doi:10.1016/j.sigpro.2011.12.012
7. Widrow, B., McCool, J.M., Larmore, M.G., Johnson, C.R.: Stationary and nonstationary learning characteristics of the lms adaptive filter. Proceedings of the IEEE 64(8), 1151–1162 (1976). doi:10.1109/PROC.1976.10286
8. Moon, T.: Session TA8a3: Adaptive Signal Processing: Theory and Applications. The Organization of information systems for government and public administration, Unesco (2010)
9. Horowitz, L., Senne, K.: Performance advantage of complex lms for controlling narrow-band adaptive arrays. IEEE Transactions on Circuits and Systems 28(6), 562–576 (1981). doi:10.1109/TCS.1981.1085024
10. Elangovan, K.: Comparative study on the channel estimation for ofdm system using lms, nllms and rls algorithms. In: International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012), pp. 359–363 (2012). doi:10.1109/ICPRIME.2012.6208372
11. Xue, Y., Zhu, X.: The minimum error entropy based robust wireless channel tracking in impulsive noise. IEEE Communications Letters 6(6), 228–230 (2002). doi:10.1109/LCOMM.2002.1010863
12. Hild, K.E., Erdogan, D., Principe, J.: Blind source separation using renyi’s mutual information. IEEE Signal Processing Letters 8(6), 174–176 (2001). doi:10.1109/97.923043
13. Radmanesh, H., Hajjabadi, M.: Recursive maximum correntropy learning algorithm with adaptive kernel size. IEEE Transactions on Circuits and Systems II: Express Briefs 65(7), 958–962 (2018). doi:10.1109/TCSII.2017.2778038
14. Wang, G., Xue, R., Wang, J.: A distributed maximum correntropy kalman filter. Signal Processing 160(Jul.), 247–251 (2019). doi:10.1016/j.sigpro.2019.02.030
15. Chen, B., Wang, J., Zhao, H., Zheng, N., Pr¨¹ncipe, J.C.: Convergence of a fixed-point algorithm under maximum correntropy criterion. IEEE Signal Processing Letters 22(10), 1723–1727 (2015). doi:10.1109/LSP.2015.2428713
16. Şen, P., Li, C.: Minimum total error entropy method for parameter estimation. IEEE Transactions on Signal Processing 63(15), 4079–4090 (2015). doi:10.1109/TSP.2015.2437836
17. Chen, B., Dang, L., Gu, Y., Zheng, N., Pr¨¹ncipe, J.C.: Minimum error entropy kalman filter. IEEE Transactions on Systems, Man, and Cybernetics: Systems 51(9), 5819–5829 (2021). doi:10.1109/TSMC.2019.2957269
18. Li, Z., Xing, L., Chen, B.: Adaptive filtering with quantized minimum error entropy criterion. Signal processing 175(Jul.) (2021). doi:10.1016/j.sigpro.2021.107534
19. Chen, B., Xing, L., Zhao, H., Zheng, N., Pr¨¹ncipe, J.C.: Generalized correntropy for robust adaptive filtering. IEEE Transactions on Signal Processing 64(13), 3376–3387 (2016). doi:10.1109/TSP.2016.2539127
20. Chen, B., Xing, L., Xu, B., Zhao, H., Zheng, N., Pr¨¹ncipe, J.C.: Kernel risk-sensitive loss: Definition, properties and application to robust adaptive filtering. IEEE Transactions on Signal Processing 65(11), 2888–2901 (2017). doi:10.1109/TSP.2017.2669903
21. Chen, B., Hu, J., Li, H., Sun, Z.: A joint stochastic gradient algorithm and its application to system identification with rbf networks. In: 2006 6th World Congress on Intelligent Control and Automation, vol. 1, pp. 1754–1758 (2006). doi:10.1109/WCICA.2006.1712654
22. Wang, G., Peng, B., Feng, Z., Yang, X., Wang, N.: Adaptive filtering based on recursive minimum error entropy criterion. Signal Processing 179 (2021). doi:10.1016/j.sigpro.2020.107836
23. Su, H., Bo, Z.: Conformal geometric algebra based band selection and classification for hyperspectral imagery. In: 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), pp. 1–4 (2016). doi:10.1109/WHISPERS.2016.8071661
24. Pham, M.T., Yoshikawa, T., Furuhashi, T., Tachibana, K.: Robust feature extractions from geometric data using geometric algebra. In: 2009 IEEE International Conference on Systems, Man and Cybernetics, pp. 529–533 (2009). doi:10.1109/ICSMC.2009.5346869
25. Wang, R., Shen, M., Wang, T., Cao, W.: L1-norm minimization for multi-dimensional signals based on geometric algebra. Advances in Applied Clifford Algebras 29(2) (2019). doi:10.1007/s00006-019-0950-7
26. Wang, R., Shen, M., Cao, W.: Multivector sparse representation for multispectral images using geometric algebra. IEEE Access 7, 12755–12767 (2019). doi:10.1109/ACCESS.2019.2892822
27. Shen, M., Wang, R., Cao, W.: Joint sparse representation model for multi-channel signal based on geometric algebra. IEEE Access 6, 24213–24223 (2018). doi:10.1109/ACCESS.2018.2819691
28. Wang, R., Wang, K., Cao, W., Wang, X.: Geometric algebra in signal and image processing: A survey. IEEE Access 7, 156315–156325 (2019). doi:10.1109/ACCESS.2019.2948615
29. Bracks, F., Schepper, N.D., Sommen, F.: The clifford-fourier transform. Journal of Fourier Analysis and Applications 6(6), 668–681 (2005). doi:10.1007/s00004-005-0479-9
30. Tao, D., Li, X., Wu, X., Maybank, S.J.: General tensor discriminant analysis and gabor features for gait recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(10), 1700–1715 (2007). doi:10.1109/TPAMI.2007.1096
31. Lopes, W.B., Al-Nuaimi, A., Lopes, C.G.: Geometric-algebra adaptive filters and its application to rotation estimation. IEEE Signal Processing Letters 23(6), 858–862 (2016). doi:10.1109/LSP.2016.2558461
32. Al-Nuaimi, A., Steinbach, E., Lopes, W.B., Lopes, C.G.: 6dof point cloud alignment using geometric algebra-based adaptive filtering. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–9 (2016). doi:10.1109/WACV.2016.7477642
33. Wang, W., Dogancay, K.: Transient performance analysis of geometric algebra least mean square adaptive filter. IEEE Transactions on Circuits and Systems II: Express Briefs PP(99), 1–1 (2021). doi:10.1109/TCSII.2021.3069390
34. Wang, W., Zhao, H., Zeng, X.: Geometric algebra correntropy: Definition and application to robust adaptive filtering. IEEE Transactions on Circuits and Systems II: Express Briefs 67(6), 1164–1168 (2020). doi:10.1109/TCSII.2019.2931507
35. Lopes, W.B., Lopes, C.G.: Geometric-algebra adaptive filters. IEEE Transactions on Signal Processing 67(14), 3649–3662 (2019). doi:10.1109/TSP.2019.2916028
36. Xie, W.X., Cao, W.M., Meng, S.: Coverage analysis for sensor networks based on clifford algebra. Science in China(Series F:Information Sciences) 51(5), 460–475 (2008). doi:10.1007/s11432-008-0048-7
37. Bayro-Corrochano, E.J., Arana-Daniel, N.: Clifford support vector machines for classification, regression, and recurrence. IEEE Transactions on Neural Networks 21(11), 1731–1746 (2010). doi:10.1109/TNN.2010.2060352
38. Wang, R., Shen, M., Wang, X., Cao, W.: Rga-cnns: convolutional neural networks based on reduced geometric
39. Hestenes, D., Sobczyk, G., Marsh, J.S.: Clifford algebra to geometric calculus. a unified language for mathematics and physics. American Journal of Physics 53(5), 510–511 (1985). doi:10.1119/1.14223

40. Wang, R., He, Y., Huang, C., Wang, X., Cao, W.: A novel least-mean kurtosis adaptive filtering algorithm based on geometric algebra. IEEE Access 7(99), 78298–78310 (2019). doi:10.1109/ACCESS.2019.2922343

41. Li, Y.F., Zheng, Z.W.: Simulation realization of adaptive filtering based on a normalized lms algorithm. Applied Mechanics and Materials 373-375, 1159–1163 (2013). doi:10.4028/www.scientific.net/AMM.373-375.1159

Figures

Figure 1 The steady-state MSD is taken as a function of kernel width and order of entropy

Figure 2 The instantaneous MSDs of the GA-MEE under various parameters

Figure 3 The instantaneous MSDs of the GA-MEE under different $\sigma$
Figure 4  The instantaneous MSDs of the GA-MEE under different $\alpha$

Figure 5  The steady-state MSD is taken as a function of kernel width and weight coefficient

Figure 6  The instantaneous MSDs of the GA-MSEMEE under various parameters

Figure 7  The instantaneous MSDs of the GA-MSEMEE under different $\sigma$

Figure 8  The instantaneous MSDs of the GA-MSEMEE under different $\eta$

Figure 9  The instantaneous MSDs of different algorithms. a GSNR=0dB; b GSNR=-1dB

Figure 10  The denoising results of 4-dimensional signal with different algorithms. a GA-LMS; b GA-NLMS; c GA-MCC; d GA-MEE; e GA-MSEMEE

Figure 11  The average 4-dimensional signal recovery errors of different algorithms

Figure 12  The denoising results of 8-dimensional signal with different algorithms. a GA-MEE; b GA-MSEMEE
Figure

Steady-state MSD vs Kernel Width and Order Of Entropy.
Figure

The graph shows the performance of five different algorithms labeled as GA-MSEME1, GA-MSEME2, GA-MSEME3, GA-MSEME4, and GA-MSEME5. The x-axis represents the number of iterations, ranging from 0 to 2000. The y-axis represents the MSD/dB, ranging from -30 to 0.

Each line on the graph corresponds to a different algorithm, with GA-MSEME1 being the most significant in terms of performance, followed by GA-MSEME2, GA-MSEME3, GA-MSEME4, and GA-MSEME5.

The performance improves over iterations, with the MSD/dB values decreasing rapidly initially and then stabilizing at lower levels for all algorithms.
Figure

Iterations

MSD

\[\eta = 7 \times 10^{-6}\]

\[\eta = 7 \times 10^{-5}\]

\[\eta = 7 \times 10^{-4}\]

\[\eta = 8 \times 10^{-4}\]

\[\eta = 9 \times 10^{-4}\]
Different algorithms

Averaged recovery error

GSNR=0dB

GSNR=-1dB

Figure
