Quaternion Kernel Normalized Minimum Error Entropy Adaptive Algorithms

Carlo Safarian* and Tokunbo Ogunfunmi†
* Department of Electrical Engineering, Santa Clara University, Santa Clara, U.S.A
E-mail: csafarian@scu.edu
† Department of Electrical Engineering, Santa Clara University, Santa Clara, U.S.A
E-mail: togunfunmi@scu.edu

***Abstract*—Information Theoretic Learning (ITL) [1][2] is gaining popularity for designing adaptive filters for a non-stationary or non-Gaussian environment. ITL cost functions such as the Minimum Error Entropy (MEE) have been applied to both linear and nonlinear adaptive filtering with better overall performance compared with the typical minimum mean squared error (MSE) and least-squares type adaptive filtering [3][4] especially for non-linear systems and in higher-order statistic noise environments. In this paper, we develop a kernel adaptive filter for quaternion data based on normalized minimum error entropy cost function. We apply generalized Hamilton-real (GHR) calculus that is applicable to Hilbert space for evaluating the cost function gradient to develop the quaternion kernel normalized minimum error entropy (QKNMEE) algorithm. The new proposed algorithm enhanced MEE algorithm where the filter update stepsize selection will be independent of the input power and the kernel size.**

I. Introduction

Quaternion valued data processing is beneficial in applications such as robotics and image processing, particularly for performing transformations in 3-dimensional space [5][6][7][13]. In particular, the benefit for quaternion valued processing includes performing data transformations in a 3 or 4-dimensional space in a more convenient fashion than using vector algebra. Transformations are performed using quaternion addition and multiplication, which differs from real or complex multiplication in that the operation is non-commutative. Applications such as pattern recognition in images and the modeling and tracking of motion are considerably simplified using quaternions [6]. Recently, [7], we applied generalized Hamilton-real (GHR) calculus that is applicable to Hilbert space [20] for evaluating the cost function gradient to develop the quaternion kernel minimum error entropy (MEE) algorithm. The MEE algorithm minimizes Renyis quadratic entropy of the error between the filter output and desired response or indirectly maximizing the error information potential. The approach improved performance for biased or non-Gaussian signals compared with the minimum mean square error criterion. While this algorithm converges quickly and has a lower misadjustment, the tracking behavior and performance has not been analyzed.

One of the main drawbacks of the MEE algorithm is its strong dependency on the kernel size σ and on the input signal power. In order to avoid these problems Han et al [4], [8] proposed an MEE based normalized algorithm, the normalized minimum error entropy (NMEE). Diniz et al addreses some of the issues from the previous works and derived a new version for the linear-in-parameter NMEE algorithm which its solution is equivalent to the privious works.

In this paper, we develop a kernel adaptive filter for quaternion data based on normalized minimum error entropy (NMEE) cost function. We explore alternative NMEE cost functions [8][14] and compare the transient, steady-state and tracking performance of the resulting algorithms with previous ones. We also present a theoretical performance analysis of the resulting algorithms and verify the transient and steady-state performance via simulations.

This paper is organized as follow: section 2 covers the background material, section 3 contains the algorithm derivation, section 4 convergence analysis, section 5 is simulation results and section 6 concludes the paper.

II. BACKGROUND

A. Quaternions and Properties

Quaternions are a 4-D associative, non-commutative, normed division algebra over the real numbers.The details about quaternions and the GHR calculus can be seen in [15][16] and [17]. Some properties of the left GHR derivatives are as follows:

$$\text{Product rule: } \frac{\partial(fg)}{\partial q^{\mu}} = f\frac{\partial g}{\partial q^{\mu}} + \frac{\partial f}{\partial q^{g\mu}}g \tag{1}$$

$$\text{Product rule: } \frac{\partial(fg)}{\partial q^{\mu*}} = f\frac{\partial g}{\partial q^{\mu*}} + \frac{\partial f}{\partial q^{g\mu*}}g \tag{2}$$

$$\text{Chain rule: } \frac{\partial(f(g(q)))}{\partial q^{\mu}} = \sum_{v\in\{1,i,j,k\}} \frac{\partial f}{\partial g^{v}}\frac{\partial g^{v}}{\partial q^{\mu}} \tag{3}$$

$$\text{Chain rule: } \frac{\partial(f(g(q)))}{\partial q^{\mu*}} = \sum_{v\in\{1,i,j,k\}} \frac{\partial f}{\partial g^{v*}}\frac{\partial g^{v*}}{\partial q^{\mu*}} \tag{4}$$

$$\text{Rotation rule: } \left(\frac{\partial f}{\partial q^{\mu}}\right)^{v} = \frac{\partial f^{v}}{\partial q^{v\mu}}, \left(\frac{\partial f}{\partial q^{\mu*}}\right)^{v} = \frac{\partial f^{v}}{\partial q^{v\mu*}} \tag{5}$$

$$\text{Conjugate rule: } \left(\frac{\partial f}{\partial q^{\mu}}\right)^* = \frac{\partial_r f^*}{\partial q^{\mu *}}, \left(\frac{\partial f}{\partial q^{\mu *}}\right)^* = \frac{\partial_r f^*}{\partial q^{\mu}} \quad (6)$$

$$\text{If f is real then: } \left(\frac{\partial f}{\partial q^{\mu}}\right)^* = \frac{\partial f}{\partial q^{\mu *}}, \left(\frac{\partial f}{\partial q^{\mu *}}\right)^* = \frac{\partial f}{\partial q^{\mu}} \quad (7)$$

B. Renyi Entropy and Parzen Window

Renyi's entropy definition such as the *order–α Renyi's entropy* is defined as [18]

$$H_{\alpha}(e) = \frac{1}{1-\alpha} \log \int_{-\infty}^{\infty} p_e^{\alpha}(e) de \quad (8)$$

where $\alpha \in \mathbb{R}^+ \backslash \{1\}$ and p_e is probability distribution function of random variable e.
We can define *order–α information potential V_{α}* as

$$V_{\alpha}(e) = \int_{-\infty}^{\infty} p_e^{\alpha}(e) de = \|p_e\|_{\alpha}^{\alpha} \quad (9)$$

where $\|.\|^{\alpha}$ is standard norm–α in L_{α}.
In practice the entropy function is not accessible since it is a function of the pdf of relative random variable e. With $\alpha = 2$ the entropy can be estimated by using some specific method such as the *Parzen window* which is a good estimation of the *order–2 Renyi's entropy* function.

For a set of N statistically independent random samples $\{e_i\}_{i=1}^{N}$ of random variable e, the *Parzen window* computes the estimate of the probability distribution function p_e as

$$\hat{p_e}(e) = \frac{1}{N\sigma} \sum_{l=1}^{N} K(\frac{e-e_l}{\sigma}) = \frac{1}{N} \sum_{l=1}^{N} G_{\sqrt{2}\sigma}(e-e_l) \quad (10)$$

where K is the real value *Gaussian Kernel* and σ is the size of kernel and $G_{\sqrt{2}\sigma}$ is defined as the following function

$$G_{\sqrt{2}\sigma}(e-e_l) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(e-e_l)^2}{2\sigma^2}\} \quad (11)$$

The estimation of information potential $\hat{V}(e)$ is given by

$$\hat{V}(e) = \frac{1}{N^2} \sum_{l_1=1}^{N} \sum_{l_2=1}^{N} G_{\sqrt{2}\sigma}(e_{l_1} - e_{l_2}) \quad (12)$$

The global solution of maximization of the $V(e)$ is the same as global solution of $\hat{V}(e)$, and with the Parzen window estimation, the global solution is achieved when all related errors are constant , i.e., $e_1 = e_2 = \; = e_N$ and the maximum value of $V(e)$ is shown by $V(0)$ or equally $\hat{V}(0) = \frac{1}{\sqrt{2}\sigma}$.

III. Quaternion Normalized Minimum Error Entropy Algorithm

In our previous work [7], we developed quaternion kernel adaptive filter based on minimum error entropy (QKMEE) with quaternion data. The goal was to maximize the information potential of the error signal. The filter could be expressed as $y_n = < \Phi(\mathbf{u}_n) , \mathbf{w}_n >$, which also can be written as:

$$y_n = \mathbf{w}_n^H \varphi_n \quad (13)$$

where both the element $\mathbf{w}_n$ and φ_n lie on a a Quaternion Reproducing Kernel Hilbert Space (QRKHS) $\mathcal{H}$ with inner product $< , >$, and $\varphi_n = \Phi(\mathbf{u}_n)$ which $\Phi(.)$ is the kernel map to a QRKHS [5].

The Normalized Minimum Error Entropy Algorithm proposed in [8] was based on the real number domain $\mathbb{R}$. In this case, we use the same method to develop Quaternion Normalized Minimum Error Entropy Algorithm (QKNMEE). The proposed parameter to be estimated in quaternion domain may be described as follow :

$$\begin{aligned} \min_{\forall \mathbf{w}_{n+1} \in \mathcal{H}} \quad & \|\mathbf{w}_{n+1} - \mathbf{w}_n\|_2^2 \\ \text{s.t.} \quad & \epsilon(n) - \epsilon(l) = 0 \\ & \forall l \in \{n-N, ...n-1\} \end{aligned} \quad (14)$$

where $\mathcal{H}$ is a quaternion RKHS and and $\epsilon(n-l) = d(n-l) - \mathbf{w}_{n+1}^H \varphi_{n-l}$ as *posteriori errors* for $\forall l : 1 \le l \le N$.

The above constrained minimization problem (14) could be converted to the following unconstrained minimization problem with cost function $J(n)$ using quaternion Lagrange multipliers $\lambda_{n-l} \in \mathbb{H}$ for $\forall l : 1 \le l \le N$:

$$J(n) = (\mathbf{w}_{n+1} - \mathbf{w}_n)^H (\mathbf{w}_{n+1} - \mathbf{w}_n) + \sum_{l=1}^{N} \lambda_{n-l} (\epsilon(n) - \epsilon(n-l)) \quad (15)$$

The minimum of $J(n)$ is reached when the gradient of $J(n)$ with respect to $\mathbf{w}_{n+1}$ is zero. The gradient of cost function $J(n)$ can be calculated in quaternion domain using GHR calculus as follow

$$\begin{aligned} \nabla_{\mathbf{w}_{n+1}^*} J(n) &= \left(\frac{\partial J(n)}{\partial \mathbf{w}_{n+1}}\right)^H \\ &= \left(\frac{\partial (\mathbf{w} - \mathbf{w}_n)^H (\mathbf{w} - \mathbf{w}_n)}{\partial \mathbf{w}_{n+1}}\right)^H \\ &+ \left(\frac{\partial \sum_{l=1}^{N} \lambda_{n-l} (\epsilon(n) - \epsilon(n-l))}{\partial \mathbf{w}_{n+1}}\right)^H \end{aligned} \quad (16)$$

where

$$\begin{aligned}\frac{\partial J(n)}{\partial \mathbf{w}_{n+1}} &= \\ &= \frac{\partial(\mathbf{w}_{n+1}^H\mathbf{w}_{n+1})}{\partial \mathbf{w}_{n+1}} - \frac{\partial(\mathbf{w}_{n+1}^H\mathbf{w}_{n})}{\partial \mathbf{w}_{n+1}} \\ &- \frac{\partial(\mathbf{w}_{n}^H\mathbf{w}_{n+1})}{\partial \mathbf{w}_{n+1}} + \frac{\partial(\mathbf{w}_{n}^H\mathbf{w}_{n})}{\partial \mathbf{w}_{n+1}} \\ &+ \sum_{l=1}^{N}\lambda_{n-l}\frac{\partial(\epsilon(n)-\epsilon(n-l))}{\partial \mathbf{w}_{n+1}}\end{aligned} \tag{17}$$

then using product rule (1) of GHR calculus the gradient can be calculated as:

$$\begin{aligned}&\frac{\partial J(n)}{\partial \mathbf{w}_{n+1}} = \\ &\mathbf{w}_{n+1}^H\frac{\partial \mathbf{w}_{n+1}}{\partial \mathbf{w}_{n+1}} + \frac{\partial \mathbf{w}_{n+1}^H}{\partial \mathbf{w}_{n+1}^{\mathbf{w}_{n+1}}}\mathbf{w}_{n+1} \\ &- \mathbf{w}_{n+1}^H\frac{\partial \mathbf{w}_{n}}{\partial \mathbf{w}_{n+1}} - \frac{\partial \mathbf{w}_{n+1}^H}{\partial \mathbf{w}_{n+1}^{\mathbf{w}_{n}}}\mathbf{w}_{n} \\ &- \mathbf{w}_{n}^H\frac{\partial \mathbf{w}_{n+1}}{\partial \mathbf{w}_{n+1}} - \frac{\partial \mathbf{w}_{n}^H}{\partial \mathbf{w}_{n+1}^{\mathbf{w}_{n+1}}}\mathbf{w}_{n+1} \\ &+ \mathbf{w}_{n}^H\frac{\partial \mathbf{w}_{n}}{\partial \mathbf{w}_{n+1}} + \frac{\partial \mathbf{w}_{n}^H}{\partial \mathbf{w}_{n}^{\mathbf{w}_{n}}}\mathbf{w}_{n} \\ &+ \sum_{l=1}^{N}\lambda_{n-l}(\frac{1}{2}\varphi_n^H - \frac{1}{2}\varphi_{n-l}^H)\end{aligned} \tag{18}$$

Using GHR calculus and derivatives properties the gradient can be simplified as:

$$\begin{aligned}&\frac{\partial J(n)}{\partial \mathbf{w}_{n+1}} \\ &= \frac{1}{2}\mathbf{w}_{n+1}^H - \frac{1}{2}\mathbf{w}_{n}^H \\ &+ \sum_{l=1}^{N}\lambda_{n-l}(\frac{1}{2}\varphi_n^H - \frac{1}{2}\varphi_{n-l}^H)\end{aligned} \tag{19}$$

therefore, by setting $\frac{\partial J(n)}{\partial \mathbf{w}_{n+1}} = 0$, the filter weight update can be calculated as :

$$\begin{aligned}\mathbf{w}_{n+1}^H =& \mathbf{w}_{n}^H - \sum_{l=1}^{N}\lambda_{n-l}(\varphi_n^H - \varphi_{n-l}^H) \\ =& \mathbf{w}_{n}^H - \Lambda\Psi_d(n)^H\end{aligned} \tag{20}$$

where $\Psi_d(n) = [\varphi_n - \varphi_{n-1}, ..., \varphi_n - \varphi_{n-N}] \in \mathcal{H}^{1\times N}$ which $\mathcal{H}$ is a quaternion RKHS and $\Lambda = [\lambda_{n-1}, ..., \lambda_{n-N}] \in \mathbb{H}^{1\times N}$ for N quaternion Lagrange multipliers.

The N Lagrange multipliers may be computed by the N constraint equations $\epsilon(n+1,n) = \epsilon(n+1,k)$ for $\forall k \in \{n-N, ...n-1\}$ using N previous posterior errors defined as $\epsilon(n,k) = d(k) - \mathbf{w}_n^H\varphi_k$. Therefore $\forall k \in \{n-N, ...n-1\}$

$$d(n) - \mathbf{w}_{n+1}^H\varphi_n = d(k) - \mathbf{w}_{n+1}^H\varphi_k \tag{21}$$

By substituting (20) in (21), we have

$$\begin{aligned}&d(n) - \Big(\mathbf{w}_n^H - \sum_{l=1}^{N}\lambda_{n-l}(\varphi_n^H - \varphi_{n-l}^H)\Big)\varphi_n \\ &= d(k) - \Big(\mathbf{w}_n^H - \sum_{l=1}^{N}\lambda_{n-l}(\varphi_n^H - \varphi_{n-l}^H)\Big)\varphi_k\end{aligned} \tag{22}$$

By using distributive property of quaternion RKHS, equation (22) can be expressed as:

$$\begin{aligned}&d(n) - \mathbf{w}_n^H\varphi_n - \sum_{l=1}^{N}\lambda_{n-l}(\varphi_n^H - \varphi_{n-l}^H)\varphi_n \\ &= d(k) - \mathbf{w}_n^H\varphi_k - \sum_{l=1}^{N}\lambda_{n-l}(\varphi_n^H - \varphi_{n-l}^H)\varphi_k\end{aligned} \tag{23}$$

By substituting the posterior errors and changing the order in equation (23), it can be simplified to:

$$\begin{aligned}&e(n) - \epsilon(n,k) = -\sum_{l=1}^{N}\lambda_{n-l}(\varphi_n^H - \varphi_{n-l}^H)(\varphi_n - \varphi_k) \\ &= -\Lambda\Psi_d(n)^H(\varphi_n - \varphi_k)\end{aligned} \tag{24}$$

Now, we define $\epsilon_d = [e(n) - \epsilon(n,n-1), ..., e(n) - \epsilon(n,n-N)] \in \mathbb{H}^{1\times N}$, and rewrite N distinct delta error equations in matrix form as:

$$\epsilon_d = -\Lambda\Psi_d(n)^H\Psi_d(n) \tag{25}$$

therefore the N quaternion Lagrange multipliers can be calculated as

$$\Lambda = -\epsilon_d\big(\Psi_d(n)^H\Psi_d(n)\big)^{-1} \tag{26}$$

where it is assumed that $\Psi_d(n)^H\Psi_d(n) \in \mathcal{H}^{N\times N}$ is non-singular and $\mathcal{H}$ is quaternion RKHS. By substituting Λ in equation (20) we can simplify filter weight update recursion formula as:

$$\mathbf{w}_{n+1}^H = \mathbf{w}_n^H + \epsilon_d\big(\Psi_d(n)^H\Psi_d(n)\big)^{-1}\Psi_d(n)^H \tag{27}$$

To simplify the weight update calculation and reduce the computational complexity due to matrix inversion, we use matrix inversion lemma and simplify the filter weight update equation (20) as:

$$\mathbf{w}_{n+1}^H = \mathbf{w}_n^H + \epsilon_d\Psi_d(n)^H\big(\Psi_d(n)\Psi_d(n)^H\big)^{-1} \tag{28}$$

or in element-wise form as:

$$\mathbf{w}_{n+1}^H = \mathbf{w}_n^H + \Bigg(\sum_{l=1}^{N}\big[e(n)-\epsilon(n-l)\big]\big[\varphi_n^H-\varphi_{n-l}^H\big]\Bigg)\times \Bigg(\big[\varphi_n-\varphi_{n-l}\big]\big[\varphi_n^H-\varphi_{n-l}^H\big]+...+ \big[\varphi_n-\varphi_{n-N}\big]\big[\varphi_n^H-\varphi_{n-N}^H\big]\Bigg)^{-1} \tag{29}$$

Using Quaternion Left Hilbert Space inner product properties we can simplify the equation (29) as:

$$\mathbf{w}_{n+1}^H = \mathbf{w}_n^H + \Bigg(\sum_{l=1}^{N}\big[e(n)-\epsilon(n-l)\big]\big[\varphi_n^H-\varphi_{n-l}^H\big]\Bigg)\times \Bigg(\big[\varphi_n^H-\varphi_{n-l}^H\big]\big[\varphi_n-\varphi_{n-l}\big]+...+ \big[\varphi_n^H-\varphi_{n-N}^H\big]\big[\varphi_n-\varphi_{n-N}\big]\Bigg)^{-1} \tag{30}$$

Therefore by expanding the vectors multiplications we can overwrite equation (30) as follow:

$$\mathbf{w}_{n+1}^H = \mathbf{w}_n^H + \Bigg(\sum_{l=1}^{N}\big[e(n)-\epsilon(n-l)\big]\big[\varphi_n^H-\varphi_{n-l}^H\big]\Bigg)\times \Bigg(\varphi_n^H\varphi_n-\varphi_n^H\varphi_{n-1}-\varphi_{n-1}^H\varphi_n+\varphi_{n-1}^H\varphi_{n-1}+... \varphi_n^H\varphi_n-\varphi_n^H\varphi_{n-N}-\varphi_{n-N}^H\varphi_n+\varphi_{n-N}^H\varphi_{n-N}\Bigg)^{-1} \tag{31}$$

Using properties of Quaternion Reproducing Kernel Hilbert Space (QRKHS) and the 'kernel trick' to replace the inner product of two vectors with quaternion kernel $\bar{\kappa}_{\bar{\sigma}}$, we can simplify the equation (31) in kernel form as:

$$\mathbf{w}_{n+1}^H = \mathbf{w}_n^H + \Bigg(\sum_{l=1}^{N}\big[e(n)-\epsilon(n-l)\big]\big[\varphi_n^H-\varphi_{n-l}^H\big]\Bigg)\times \Bigg(\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_n},\mathbf{u_n})-2\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_n},\mathbf{u_{n-1}})+\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{n-1}},\mathbf{u_{n-1}})+... \bar{\kappa}_{\bar{\sigma}}(\mathbf{u_n},\mathbf{u_n})-2\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_n},\mathbf{u_{n-N}})+\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{n-N}},\mathbf{u_{n-N}})...\Bigg)^{-1} \tag{32}$$

For use of the kernel trick, we use quaternion-extended real Gaussian kernel in [5], where $\bar{\kappa}_{\bar{\sigma}}(.,.)$ is real Gaussian kernel. Therefore in equation (32) the inverse term is real number can be moved to right or left side in multiplication with quaternion numbers. By setting $\mathbf{w}_0^H=0$ and including a step size factor η, the weight update recursion can be calculated as:

$$\mathbf{w}_n^H = \eta\sum_{p=0}^{n-1}\Bigg(\sum_{l=1}^{N}\big[e(p)-\epsilon(p-l)\big]\big[\varphi_p^H-\varphi_{p-l}^H\big]\Bigg)\times \Bigg(\sum_{l=1}^{N}\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_p})-2\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_{p-l}})+\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{p-l}},\mathbf{u_{p-l}})\Bigg)^{-1} \tag{33}$$

By substituting the weight update in the $y_n = \mathbf{w}_n^H\varphi_n$ and using properties of Quaternion Reproducing Kernel Hilbert Space (QRKHS) and the 'kernel trick' to replace the inner product of two vectors with quaternion kernel $\bar{\kappa}_{\bar{\sigma}}$, equation (13) can be simplified in kernel form as:

$$y_n = \eta\sum_{p=0}^{n-1}\Bigg(\sum_{l=1}^{N}\big[e(p)-\epsilon(p-l)\big]\big[\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_n})-\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{p-1}},\mathbf{u_n})\big]\Bigg) \times\Bigg(\sum_{l=1}^{N}\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_p})-2\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_{p-l}})+\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{p-1}},\mathbf{u_{p-l}})\Bigg)^{-1} \tag{34}$$

IV. Convergence Analysis

The goal of the convergence analysis is to find a range for learning step size η in equation (33) which QKNMEE converges to optimal set of weights. To studying convergence of QKMEE algorithm, we consider an approach using the energy conservation relation [19]. The weight error at iteration $n+1$ can be defined as:

$$\begin{aligned}\mathbf{v}_{n+1} &= \mathbf{w}^0-\mathbf{w}_{n+1}\\ &= \mathbf{w}^0-(\mathbf{w}_n+\Delta\mathbf{w}_n)\\ &= \mathbf{v}_n-\Delta\mathbf{w}_n\end{aligned} \tag{35}$$

For checking energy conservation, we initially find prior and posteriori errors: $e_n^a = \mathbf{v}_n^H\varphi_n$ and $e_n^p = \mathbf{v}_{n+1}^H\varphi_n$ where

$$\begin{aligned}e_n^p &= \mathbf{v}_{n+1}^H\varphi_n\\ &= (\mathbf{v}_n^H-\Delta\mathbf{w}_n^H)\varphi_n\\ &= e_n^a-\Delta\mathbf{w}_n^H\varphi_n\\ &= e_n^a-\eta\\ &\times\Bigg(\sum_{l=1}^{N}\big[e(n)-e(n-l)\big]\big[\varphi_n^H\varphi_n-\varphi_{n-l}^H\varphi_n\big]\Bigg)\\ &\times\Bigg(\sum_{l=1}^{N}\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_p})-2\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_{p-l}})+\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{p-1}},\mathbf{u_{p-l}})\Bigg)^{-1}\end{aligned} \tag{36}$$

To simplify calculation, function $\gamma(n)$ can be defined as:

$$\gamma(n) \triangleq \Bigg(\sum_{l=1}^{N}\big[e(n)-e(n-l)\big]\big[\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_n},\mathbf{u_n})-\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{n-1}},\mathbf{u_n})\big]\Bigg) \times\Bigg(\sum_{l=1}^{N}\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_p})-2\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_p},\mathbf{u_{p-l}})+\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{p-1}},\mathbf{u_{p-l}})\Bigg)^{-1} \tag{37}$$

thus, the energy can be expressed as:

$$\begin{aligned}\left\|\mathbf{v}_{n+1}^H\varphi_n\right\|^2 &= \left\|\mathbf{v}_n^H\varphi_n - \eta\gamma(n)\right\|^2 \\ &= \left\|\mathbf{v}_n^H\varphi_n\right\|^2 + \mathbf{v}_n^H\varphi_n\big(-\eta\gamma^*(n)\big) \\ &+ \big(-\eta\gamma(n)\big)\big(\mathbf{v}_n^H\varphi_n\big)^* \\ &+ \left\|-\eta\gamma(n)\right\|^2 \\ &= \left\|\mathbf{v}_n^H\varphi_n\right\|^2 - 2\eta\mathfrak{Re}\big(\mathbf{v}_n^H\varphi_n\gamma^*(n)\big) \\ &+ \eta^2\left\|\gamma(n)\right\|^2\end{aligned} \tag{38}$$

Using Cauchy Schwarz inequality in Hilbert space and normalized kernel $\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_n}, \mathbf{u_n}) = 1$, we can expressed the following inequality as:

$$\left\|\mathbf{v}_{n+1}^H\varphi_n\right\|^2 \le \|\mathbf{v}_{n+1}\|^2\|\varphi_n\|^2 = \|\mathbf{v}_{n+1}\|^2\,\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_n}, \mathbf{u_n}) \tag{39}$$

therefore

$$\left\|\mathbf{v}_{n+1}^H\varphi_n\right\|^2 \le \|\mathbf{v}_{n+1}\|^2 \tag{40}$$

and

$$\left\|\mathbf{v}_n^H\varphi_n\right\|^2 \le \|\mathbf{v}_n\|^2 \tag{41}$$

by subtracting (41) form (40), inequality can be written as:

$$\left\|\mathbf{v}_{n+1}^H\varphi_n\right\|^2 - \left\|\mathbf{v}_n^H\varphi_n\right\|^2 \le \|\mathbf{v}_{n+1}\|^2 - \|\mathbf{v}_n\|^2 \tag{42}$$

by taking expectation of both sides of (42) :

$$E\left\|\mathbf{v}_{n+1}^H\varphi_n\right\|^2 - E\left\|\mathbf{v}_n^H\varphi_n\right\|^2 \le E\|\mathbf{v}_{n+1}\|^2 - E\|\mathbf{v}_n\|^2 \tag{43}$$

For convergence, the energy of the weight error vector should gradually reduce per iteration. Thus

$$E\left[\left\|\mathbf{v}_{n+1}^H\varphi_n\right\|^2\right] - E\left[\left\|\mathbf{v}_n^H\varphi_n\right\|^2\right] < 0 \tag{44}$$

therefore, by taking expectation of both sides of equation (38) and using inequality (44)

$$\begin{aligned}&E\left[\left\|\mathbf{v}_{n+1}^H\varphi_n\right\|^2\right] - E\left[\left\|\mathbf{v}_n^H\varphi_n\right\|^2\right] \\ &= -2\eta E\left[\mathfrak{Re}\big(\mathbf{v}_n^H\varphi_n\gamma^*(n)\big)\right] \\ &+ \eta^2 E\left[\left\|\gamma(n)\right\|^2\right] \\ &< 0\end{aligned} \tag{45}$$

Thus, in order the algorithm converges, the convergence step size η should be:

$$\eta < 2\frac{E\left[\mathfrak{Re}\big(e_n^p\gamma^*(n)\big)\right]}{E\left[\left\|\gamma(n)\right\|^2\right]} \tag{46}$$

V. SIMULATION RESULTS

A. channel estimation based on the Weiner nonlinear model

The Quat-KNMEE (QKNMEE) algorithm was simulated with Parzen Window lenght $N = 10$ for a nonlinear channel with non-Gaussian noise versus Quat-KLMS[13]. The channel consisted of the quaternion filter, i.e.,

$$\begin{aligned}&z(n) = g_1^*u(n) + g_2^*u^i(n) + g_3^*u^j(n) + g_4^*u^k(n) \\ &+h_1^*u(n-1) + h_2^*u^i(n-1) + h_3^*u^j(n-1) + h_4^*u^k(n-1)\end{aligned}$$

and nonlineraity, i.e.,

$$y(n) = z(n) + az^2(n) + bz^3(n) + v(n)$$

where $v(n)$ is added non-Gaussian noise described later. Coefficients $g1, ..., g4, h1, ..., h4, a, b$, and noise $v(n)$ are all quaternion valued.The coefficients used were

$$\begin{aligned}&a = 0.075 + i0.35 + j0.1 - k0.05, \\ &b = -0.025 - i0.25 - j0.05 + k0.03, \\ &g1 = -0.40 + i0.30 + j0.15 - k0.45, \\ &h1 = 0.175 - i0.025 + j0.1 + k0.15, \\ &g2 = -0.35 - i0.15 - j0.05 + k0.20, \\ &h2 = 0.15 - i0.225 + j0.125 - k0.075, \\ &g3 = -0.10 - i0.40 + j0.20 - k0.05, \\ &h3 = +0.025 + i0.075 - j0.05 - k0.05, \\ &g4 = +0.35 + i0.10 - j0.10 - k0.15, \\ &h4 = -0.05 - i0.075 - j0.075 + k0.175.\end{aligned}$$

For the tests, both input $u(n)$ and noise $v(n)$ were formed using impulsive Gaussian mixture models to form non-Gaussian signals. A quaternion random variable with components from different real Gaussian distributions was formed [7]. The probability distributions used were

$$\begin{aligned}&p_u(i) = (0.85N(1.0, 0.01) + 0.15N(3.0, 0.01)) \\ &+ i(0.40N(0.5, 0.01) + 0.60N(2.5, 0.01)) \\ &+ j(0.65N(3.5, 0.01) + 0.35N(1.5, 0.01)) \\ &+ k(0.25N(2.0, 0.01) + 0.75N(5.5, 0.01))\end{aligned}$$

$$\begin{aligned}&p_v(i) = (0.90N(0.0, 0.01) + 0.10N(1.0, 0.01)) \\ &+ i(0.70N(3.0, 0.01) + 0.30N(0.5, 0.01)) \\ &+ j(0.45N(1.0, 0.01) + 0.55N(4.5, 0.01)) \\ &+ k(0.80N(0.5, 0.01) + 0.20N(1.5, 0.01))\end{aligned}$$

where $N(m_N, \sigma_N)$ denotes the normal (Gaussian) PDF with mean m_N and variance σ_N. The Quat-KNMEE and Quat-KLMS simulation results for the nonlinear channel described are shown in Fig. 1 to Fig 2. Figs. 1 which are ensemble-averaged over 10 realizations.

To compare the performance of new proposed algorithm Quat-KNMEE with Quat-KMEE and Quat-KLMS, the parameters of all three algorithms were chosen that all three algorithms reached the same steady state mean square errors. For this reason the parameters for the Quat-KNMEE were

$\eta = 3$, $\bar{\sigma} = 2.24$ and for the Quat-KMEE were $\eta = 0.4$, $\bar{\sigma} = 2.24$, and $\sigma = 0.736$ and for the Quat-KLMS $\eta = 0.5$, $\bar{\sigma} = 2.24$ were used.

Fig. 1 shows the performance comparisons when the input power was set to -5.1 dB and measurement noise = 12.5 dBm. As shown in Fig. 1, The new proposed algorithm Quat-KNMEE converged with 1000 iterations where the Quat-KMEE converged with 2000 iterations. It is clear from Fig 1, that the new proposed algorithm Quat-KNMEE converges faster compared to the other two algorithms Quat-KMEE and Quat-KLMS.

To show that the Quat-KNMEE filter update stepsize selection is independent of the input power, the input power of second simulation was increased to 1.75dB while all stepsize of all three algorithms were kept the same as before. Fig. 2 shows the input power impacts on three algorithms when input power was set to 1.75dB. As shown in Fig. 2, the convergence rate of the Quat-KNMEE didn't change and converged within 1000 iterations while the convergence rate and stability of the other two algorithms Quat-KMEE and Quat-KLMS changed and converged faster compare to the first simulation using smaller input power.

Fig. 3 shows the learning curves of the Quat-KNMEE filter with different convergence step sizes. As shown in Fig. 3 when the step size parameter η increases, the rate of convergence of Quat-KNMEE algorithm is correspondingly increased. When the step size is set to values greater than 5, the algorithm couldn't converge and becomes unstable.

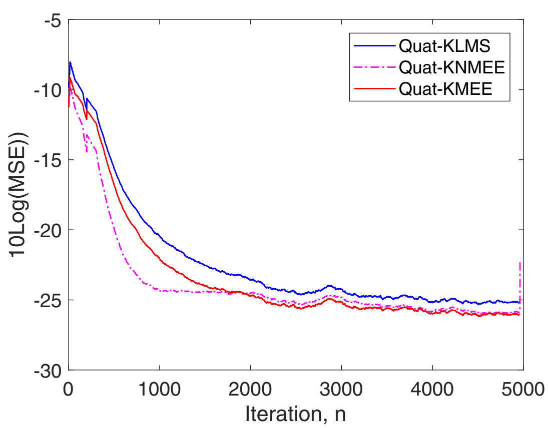

Fig. 1. Learning curves for Quat-KNMEE, Quat-KMEE and Quat-KLMS and for non-Gaussian signal with input power = -5.1 dB

VI. Conclusion

We have shown the derivation and convergence analysis of a quaternion kernel adaptive algorithm based on normalized minimum error entropy. The algorithm is based on information theoretic learning (ITL) cost function. The resulting algorithm is the Quat-KNMEE algorithm using GHR calculus. A gradient is derived based on quaternion RKHS. Simulation

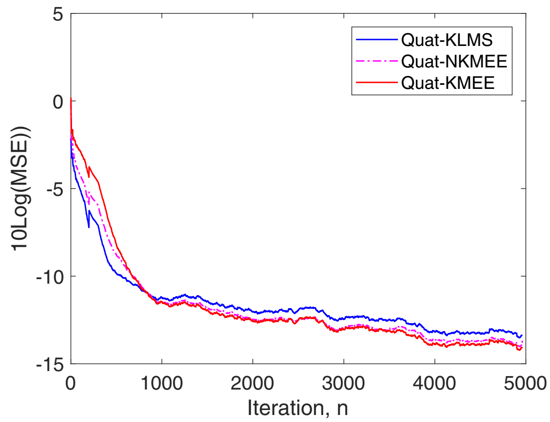

Fig. 2. Learning curves for Quat-KNMEE, Quat-KMEE and Quat-KLMS and for non-Gaussian signal with input power = 1.75 dB

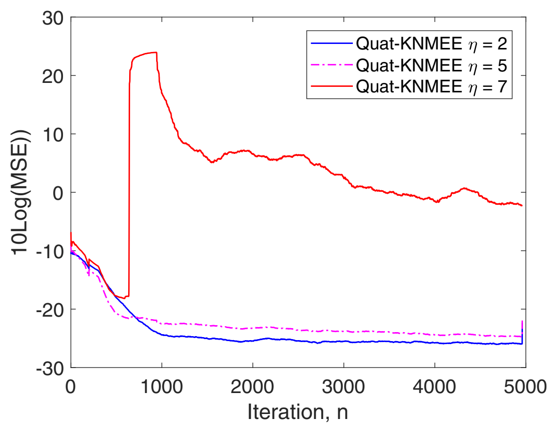

Fig. 3. Learning curves for Quat-KNMEE with different convergence step size for non-Gaussian signal with input power = -5.1 dB

results show the convergence curve of the mean square error of the new algorithm (QKNMEE) versus the existing algorithms Quat-KMEE (QKMEE) and Quat-KLMS (QKLMS). The algorithm's convergence is very fast and outperforms the existing one QKMEE and QKLMS. The convergence analysis (46) shows that convergence step-size is independent of kernel size. The simulation results show that the convergence rate of the Quat-KNMEE is independent of the input power and the kernel size. QKNMEE algorithm gives better performance for low signal to noise ratio (SNR) environments.

References

[1] D. Erdogmus and J.C. Principe, "From linear adaptive filter to nonlinear information processing," *IEEE Signal Processing Magazine,* Vol. 23, No. 6, pp. 14-33, Nov. 2006.

[2] W. Liu, I. M. Park, and J. C. Principe, "An information theoretic approach of designing sparse kernel adaptive filters," *IIEEE Transactions on Neural Networks* , Volume: 20, Issue: 12, Dec. 2009.

[3] Tokunbo Ogunfunmi, "Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches," *published by Springer Publishers,* 2007.
[4] David T. Westwick and Robert E. Kearney, "Identification of Nonlinear Physiological Systems," *IEEE/Wiley Interscience Publishers,* 2003.
[5] T. K. Paul, and T. Ogunfunmi, "A kernel adaptive algorithm for quaternion-valued inputs," *IEEE Transactions on Neural Networks and Learning Systems,* Volume: 26, Issue: 10, Oct. 2015.
[6] W. Liu, J. C. Principe, and S. Haykin, "Kernel Adaptive Filtering : A Comprehensive Introduction," *published by Wiley Publishers,* 2010.
[7] Tokunbo Ogunfunmi and Carlo Safarian, "A Quaternion Kernel Minimum Error Entropy Adaptive Filter," *Proceedings of the IEEE ICASSP,* 2018.
[8] Wallace A. Martins, Paulo S. R. Diniz, Yih-Fang Huang, "On the Normalized Minimum Error Entropy Adaptive Algorithm: Cost Function and Update Recursion," *Circuits and Systems (LASCAS), 2010 First IEEE Latin American Symposium on,* Feb. 2010.
[9] D. P. Mandic and V. S. L. Goh, "Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models," *New York: Wiley,* 2009.
[10] A. Hirose, "Complex-valued neural networks," *Springer-Verlag, Berlin,* 2006.
[11] Tokunbo Ogunfunmi and Thomas Paul, "On the Complex Kernel-based Adaptive Filter," *Proceedings of the IEEE ISCAS,* pp. 1263-1266, May 2011.
[12] Tokunbo Ogunfunmi and Thomas Paul, "Kernel-Based APA Adaptive Filters for Complex Data," *Proceedings of the Asia Pacific Signal and Information Processing. Association (APSIPA) Conference,* Oct. 2011.
[13] Tokunbo Ogunfunmi and Thomas Paul, "An Alternative Kernel Adaptive Filtering Algorithm for Quaternion Data," *Proceedings of the Asia Pacific Signal and Information Processing. Association,* Dec. 2012.
[14] A. H. Sayed, "Fundamentals of Adaptive Filtering," *New York: Wiley-Interscience,* 2003.
[15] Dongpo Xu, Cyrus Jahanchahi, Clive C. Took and Danilo P. Mandic, "Quaternion Derivatives: The GHR Calculus," *Submitted exclusively to the London Mathematical Society doi:10.1112/0000/000000,* 2014.
[16] Dongpo Xu, Yili Xia and Danilo P. Mandic, "Optimization in Quaternion Dynamic Systems:Gradient, Hessian, and Learning Algorithms," *IEEE Transactions on Neural Networks and Learning Systems,* VOL. 27, NO. 2, pp. 249 - 261, 2016.
[17] M. D. Jiang, Y. Li and W. Liu, "Properties of a General Quaternion-Valued Gradient Operator and Its Application to Signal Processing," *Frontiers of Information Technology and Electronic Engineering,* vol. 17, issue:2, pp. 83-95, 2016
[18] Deniz Erdogmus, Jose C. Principeg, "An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems," *IEEE Transactions on Signal Processing,* Volume: 50, Issue: 7, pp. 1780 - 1786, 2012
[19] S. Zhao, B. Chen, and J. C. Principe, "Kernel Adaptive Filtering with Maximum Correntropy Criterion," *IEEE The 2011 International Joint Conference on Neural Networks,* 2011,pp. 2012 - 2017
[20] Felipe A. Tobar and Danilo P. Mandic, "Quaternion Reproducing Kernel Hilbert Spaces:Existence and Uniqueness Conditions," *IEEE Transactions on Information Theory,* 2014, Volume: 60, Issue: 9, pp. 5736 - 5749.