A Quaternion Kernel Minimum Error Entropy Adaptive Filter

Tokunbo Ogunfunmi

Carlo Safarian

Signal Processing Research Lab (SPRL),

Department of Electrical Engineering,

Santa Clara University, CA 95053, USA

Friday , April 20, 2018

Presented at the IEEE ICASSP 2018

Santa Clara University

School of Engineering

Outline

– In this talk, we will cover the following:

- Motivation
- Review of Kernel Methods
- Quaternions and GHR Calculus
  - Quaternion Adaptive Filtering
- Renyi's Entropy and Parzen window
- Quaternions and Kernel Filtering using Minimum Entropy cost function
  - Kernel-based Estimation with Quaternions minimum entropy
  - Derivation of Quaternion KMEE
- Simulations
- Conclusions

Motivation

– Develop Adaptive filter for nonlinear system for Quaternion data, based on information theoretic learning cost function.

- Use Kernel method for nonlinearity and computational complexity
- Use Information Theoretic Learning (ITL) cost function such as Error Entropy.
  - Entropy is the average information of a random variable.
- Use Minimum Error Entropy Cost Function:
  - By minimizing error entropy, the amount of information lost to the error signal is minimized.
- Use GHR calculus to calculate the Gradient of cost function in Quaternion domain.

Kernel-Based Algorithms

- Kernel methods allow for recasting a **nonlinear optimization** problem to a space where **linear optimization** can be used.

- **Kernel-based methods** provide a powerful approach for performing nonlinear adaptive filtering.

- In nonlinear system filtering using **kernel methods** can greatly **reduce the complexity** of equalization or estimation.

Kernel-Based Algorithms

– The **kernel** transform the input to ***higher-dimension*** space, referred to as the **feature space** using **feature map**.

- Example:

  For input $\mathbf{x}=[x_1\ x_2]$, a possible **feature space** mapping is:

  $$\boldsymbol{\Phi}(\mathbf{x}) = [\,x_1 \quad x_2 \quad x_1^2 \quad x_1x_2 \quad x_2^2\,] \quad \text{(for order 2 nonlinearities)}$$

  where $\boldsymbol{\Phi}(.)$ is the **transform map** or **feature map**.

- The transform shown allows **adaptive learning** of a **nonlinear channel** of order 2 to be **performed linearly** in feature space.

Kernel-Based Algorithms

– Kernel functions

- compute the **inner product** of two input vectors in **feature space**, and reduce the computation complexity :

$$\kappa(\mathbf{x}, \mathbf{y}) \;=\; <\boldsymbol{\Phi}(\mathbf{x}), \boldsymbol{\Phi}(\mathbf{y})>_S$$

- Example of Kernel: Gaussian kernel:

$$\kappa(\mathbf{x}, \mathbf{y}) \;=\; \exp(-\sigma|\mathbf{x} - \mathbf{y}|^2), \qquad \text{where } \sigma \text{ adjusts mapping}$$

- This kernel represents the inner product for an **infinite-dimensional** space. Thus, the Gaussian kernel allows for learning in an infinite-dimension space with **finite complexity.**
- Use of **infinite-dimension spaces** allow for **universal function** approximation (i.e. learning nonlinearities of any order).

– Kernel trick

- If an algorithm can be formed with **inner products** or equivalent kernel evaluation using **kernel functions** there is no need to perform computation in higher dimension, this is referred to as the '**kernel trick**'.

Quaternions

– The use of quaternion-valued data has been drawing recent interest in various areas of statistical signal processing [1]:

- Vector sensors such as motion body sensors, seismic, wind modeling
- Adaptive filters
- Machine learning

– The benefit for quaternion-valued processing in particular includes performing data transformations in 3 or 4-dimensional space conveniently compared to vector algebra.

[1] C. C. Took and D. P. Mandic, "A quaternion widely linear adaptive filter," in Trans. Signal Process. IEEE, 2010, vol. 58, no. 8, pp. 44274431.

Quaternions

- **Quaternions** are **4-dimensional** data values, composed of a real component and **3 orthonormal** imaginary components: i, j, and k

$$q = q_a + iq_b + jq_c + kq_d,$$

where q_a = real component
q_b = imag. comp. along i
q_c = imag. comp. along j
q_d = imag. comp. along k

- **Quaternions** are **noncommutative** in **multiplication**: $q_1 q_2 \neq q_2 q_1$

- Multiplication of quaternions is based on the following properties. For the imaginary components: $ij = -ji = k,\quad jk = -kj = i,\quad ki = -ik = j,$

- An important relation for quaternion data are **quaternion involutions**. These relations are:

$$\begin{aligned} q^i &= -iqi &&= q_a + iq_b - jq_c - kq_d \\ q^j &= -jqj &&= q_a - iq_b + jq_c - kq_d \\ q^k &= -kqk &&= q_a - iq_b - jq_c + kq_d \end{aligned}$$

HR Derivatives

– The **HR derivatives** are:

$$\begin{pmatrix} \frac{\partial f}{\partial q} \\ \frac{\partial f}{\partial q^i} \\ \frac{\partial f}{\partial q^j} \\ \frac{\partial f}{\partial q^k} \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 & -i & -j & -k \\ 1 & -i & j & k \\ 1 & i & -j & -k \\ 1 & i & j & -k \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial q_r} \\ \frac{\partial f}{\partial q_i} \\ \frac{\partial f}{\partial q_j} \\ \frac{\partial f}{\partial q_k} \end{pmatrix}$$

– The conjugate (**HR* derivatives**):

$$\begin{pmatrix} \frac{\partial f}{\partial q^*} \\ \frac{\partial f}{\partial q^{i*}} \\ \frac{\partial f}{\partial q^{j*}} \\ \frac{\partial f}{\partial q^{k*}} \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 & i & j & k \\ 1 & i & -j & -k \\ 1 & -i & j & -k \\ 1 & -i & -j & k \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial q_r} \\ \frac{\partial f}{\partial q_i} \\ \frac{\partial f}{\partial q_j} \\ \frac{\partial f}{\partial q_k} \end{pmatrix}$$

- HR calculus can be difficult in complex optimization problems due to the lack of product and chain rules, a consequence of the non-commutativity of quaternion algebra. Generalized HR (GHR) derivatives leverages the quaternion rotations in a general orthogonal system, provide the left- and right-hand versions of the quaternion derivative of general functions.

[6] Dongpo Xu, Cyrus Jahanchahi, Clive C. Took and Danilo P. Mandic, "Quaternion Derivatives: The GHR Calculus," in Submitted exclusively to the London Mathematical Society doi:10.1112/0000/000000. 2014Paper about Gradients of Quaternion
[7] Dongpo Xu, Yili Xia, Member, IEEE, and Danilo P. Mandic, "Optimization in Quaternion Dynamic Systems: Gradient, Hessian, and Learning Algorithms," in Transactions on Neural Networks and Learning Systems. IEEE,2016, VOL. 27, NO. 2, pp. 249 - 261.
[16] M. D. Jiang, Y. Li, and W. Liu , "Properties of a General Quaternion-Valued Gradient Operator and Its Application to Signal Processing," in Frontiers of Information Technology and Electronic Engineering, 2016, vol. 17, issue: 2, pp. 83-95 .

GHR Derivatives

- The **GHR derivatives** (Left) :

$$\frac{\partial f}{\partial q^{\mu}} = \frac{1}{4}\left(\frac{\partial f}{\partial q_r} - \frac{\partial f}{\partial q_i} i^{\mu} - \frac{\partial f}{\partial q_j} j^{\mu} - \frac{\partial f}{\partial q_k} k^{\mu}\right)$$

- The conjugate **GHR* derivatives** (Left) are:

$$\frac{\partial f}{\partial q^{\mu *}} = \frac{1}{4}\left(\frac{\partial f}{\partial q_r} + \frac{\partial f}{\partial q_i} i^{\mu} + \frac{\partial f}{\partial q_j} j^{\mu} + \frac{\partial f}{\partial q_k} k^{\mu}\right)$$

- While the right side derivatives are :

$$\frac{\partial_r f}{\partial q^{\mu}} = \frac{1}{4}\left(\frac{\partial f}{\partial q_r} - i^{\mu}\frac{\partial f}{\partial q_i} - j^{\mu}\frac{\partial f}{\partial q_j} - k^{\mu}\frac{\partial f}{\partial q_k}\right)$$

- While the right side derivatives are :

$$\frac{\partial_r f}{\partial q^{\mu *}} = \frac{1}{4}\left(\frac{\partial f}{\partial q_r} + i^{\mu}\frac{\partial f}{\partial q_i} + j^{\mu}\frac{\partial f}{\partial q_j} + k^{\mu}\frac{\partial f}{\partial q_k}\right)$$

[6] Dongpo Xu, Cyrus Jahanchahi, Clive C. Took and Danilo P. Mandic, "Quaternion Derivatives: The GHR Calculus," in Submitted exclusively to the London Mathematical Society doi:10.1112/0000/000000. 2014Paper about Gradients of Quaternion
[7] Dongpo Xu, Yili Xia, Member, IEEE, and Danilo P. Mandic, "Optimization in Quaternion Dynamic Systems: Gradient, Hessian, and Learning Algorithms," in Transactions on Neural Networks and Learning Systems. IEEE,2016, VOL. 27, NO. 2, pp. 249 - 261.

GHR Derivatives Properties

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

$$\frac{\partial(fg)}{\partial q^{\mu *}} = f\frac{\partial g}{\partial q^{\mu *}} + \frac{\partial f}{\partial q^{g\mu *}}g$$

– Chain rule:

$$\frac{\partial(f(g(q)))}{\partial q^{\mu}} = \sum_{v\in\{1,i,j,k\}} \frac{\partial f}{\partial g^{v}}\frac{\partial g^{v}}{\partial q^{\mu}}$$

$$\frac{\partial(f(g(q)))}{\partial q^{\mu *}} = \sum_{v\in\{1,i,j,k\}} \frac{\partial f}{\partial g^{v *}}\frac{\partial g^{v *}}{\partial q^{\mu *}}$$

– Rotation rule:

$$\left(\frac{\partial f}{\partial q^{\mu}}\right)^{v} = \frac{\partial f^{v}}{\partial q^{v\mu}}, \left(\frac{\partial f}{\partial q^{\mu *}}\right)^{v} = \frac{\partial f^{v}}{\partial q^{v\mu *}}$$

– Conjugate rule:

$$\left(\frac{\partial f}{\partial q^{\mu}}\right)^{*} = \frac{\partial_r f^{*}}{\partial q^{\mu *}}, \left(\frac{\partial f}{\partial q^{\mu *}}\right)^{*} = \frac{\partial_r f^{*}}{\partial q^{\mu}}$$

Gaussian-based kernel for quaternion data

– Gaussian-based kernel for quaternion data may be expressed as:

$$\kappa_\sigma(X-Y) = \frac{4}{\sqrt{2\pi\sigma}}\exp\{\frac{-1}{2\sigma^2}(X_r-Y_r)^2+(X_i-Y_i)^2+ (X_j-Y_j)^2+(X_k-Y_k)^2\}$$
$$= \frac{4}{\sqrt{2\pi\sigma}}\exp\{\frac{-1}{2\sigma^2}|X-Y|^2\}$$

– where X and Y are quaternion numbers in $\mathbf{H}$ in form of

– $X = X_r + iX_i - jX_j - kX_k$ and $Y = Y_r + iY_i - jY_j - kY_k$

– More details of the quaternion kernel is provided in [15].

[15] Thomas K. Paul and Tokunbo Ogunfunmi , "A Kernel Adaptive Algorithm for Quaternion-Valued Inputs," in Transactions on Neural Networks and Learning Systems.IEEE, 2015,Volume: 26, Issue: 10,pp. 2422 - 2439

Information Theoretic Learning Cost Functions

- MCC (maximum correntropy criterion)
  - It is a local criterion because it only cares about the local part of the error PDF falling within the kernel bandwidth. When the error modes are far from the origin, they fall outside the kernel bandwidth, and the learning stalls.
  - Lower computational complexity
- MEE (minimum error entropy)
  - It weights the error PDF by itself, the error modes are easily detected with the advantage of data efficiency and more effective training.
  - Higher computational complexity

[17] Weifeng Liu, P. P. Pokharel, J. C. Principe , “Error Entropy, Correntropy and M-Estimation,” in 2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing. IEEE, 2006,pp. 179 - 184 .

Minimum Error Entropy

– Why use Minimum Error Entropy:

- Entropy is a scalar quantity that provides a measure for the average information contained in a given probability distribution function.
- By definition, information is a function of the pdf; hence, entropy as an optimality criterion extends MSE. [13]
- When the error entropy is minimized the amount of information lost to the error signal minimized.
- In order to extract the most information from the data in supervised learning, given samples from an input-output mapping, the information content of the error signal must be minimized, therefore the error entropy over the training data set must be minimized. Then, all moments of the error pdf (not only the second moments) are constrained.

[13] Deniz Erdogmus; Jose C. Principe, "An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems," in Transactions on Signal Processing. IEEE, 2002, Volume: 50, Issue: 7, pp. 1780 - 1786.

Renyi’s Entropy

– Why use Renyi’s Entropy

- Minimizing Renyi's error entropy minimizes the Renyi's divergence between the joint pdfs of the input-desired signals and the input-output signals[13].

- Reyni’s entropy definition such as the *order-α* Renyi’s entropy is defined as :

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

  – Where $\alpha \in \mathbb{R}^+ \backslash \{1\}$ and p_e distribution function of random variable ***e***

- We can define *order-α information potential Vα* :

$$V_\alpha(e) = \int_{-\infty}^{\infty} p_e^\alpha(e)de = \|p_e\|_\alpha^\alpha \qquad \|.\|^\alpha \text{ is standard norm–}\alpha \text{ in } L_\alpha$$

[13] Deniz Erdogmus; Jose C. Principe, “An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems,” in Transactions on Signal Processing. IEEE, 2002, Volume: 50, Issue: 7, pp. 1780 - 1786.

Parzen Window

– Why use Parzen Window

- 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 *Parzen window* which is a good estimation of the *order-2 Renyi's* 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: $\hat{p_e}(e) = \frac{1}{N\sigma}\sum_{l=1}^{N} K(\frac{e-e_l}{\sigma}) = \frac{1}{N}\sum_{l=1}^{N} G_{\sqrt{2}\sigma}(e-e_l)$

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

Santa Clara University
School of Engineering

Estimation of Information Potential *V(e)*

– The estimation of *information potential V(e)* using *Parzen window* is:

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

– The global solution of maximization of the $V(e)$ is the same as global solution of $\hat{V}(e)$ with the *Parzen window* estimation and the global solution is achieved when all related errors are constant.

– The maximum value of $V(e)$ is :

$$V(0) = \hat{V}(0) = \frac{1}{\sqrt{2}\sigma}$$

Minimizing the Error Entropy

- The Minimizing the error entropy can be done by maximizing the error information potential cost function *J (n)* in quaternion domain which can be defined as :

$$J(n) = \frac{1}{N^2} \sum_{i,j=1}^{N} G_{q,\sqrt{2}\sigma}(e(n-i) - e(n-j))$$

- Where

$$G_{q,\sqrt{2}\sigma}(e_i - e_j) = \frac{4}{\sqrt{2\pi}\sigma} \exp\{-\frac{\left|e_i - e_j\right|^2}{2\sigma^2}\}$$

$$e_i \text{ and } e_j \in \mathbb{H}$$

Santa Clara University
School of Engineering

Quaternion Minimum Error Entropy Algorithm

Quaternion Minimum Error Entropy Algorithm

- For the quaternion kernel adaptive filter based on minimum entropy (QKMEE) with quaternion data, the goal is to minimize the error entropy which can be done with maximizing the information potential cost function *J(n)*.

- The filter can be expressed as:

$$y_n = < \Phi(\mathbf{u}_n) \ , \mathbf{w}_n >= \mathbf{w}_n^H \varphi_n$$

- Where $\varphi_n = \Phi(\mathbf{u}_n)$ and $\Phi(.)$ is the kernel map to a quaternion RKHS [15]

Santa Clara University
School of Engineering

[15] Thomas K. Paul and Tokunbo Ogunfunmi, "A Kernel Adaptive Algorithm for Quaternion-Valued Inputs," in Transactions on Neural Networks and Learning Systems.IEEE, 2015,Volume: 26, Issue: 10,pp. 2422 - 2439

Quaternion Minimum Error Entropy Algorithm

– Maximizing the information potential cost function *J(n)* can be done with unconstrained optimization algorithm such as gradient ascent algorithm.

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \eta \nabla_{\mathbf{w}_n^*} J(n) = \mathbf{w}_n + \eta \left(\frac{\partial J(n)}{\partial \mathbf{w}_n} \right)^H$$

$$= \mathbf{w}_n + \mu \left(\frac{\partial \left[\sum_{l,t=1}^{N} \exp\{ -\frac{|e(n-l)-e(n-t)|^2}{2\sigma^2} \} \right]}{\partial \mathbf{w}_n} \right)^H$$

– Where η is adaptation step size and $\mu = \eta \frac{1}{N^2} \frac{4}{\sqrt{2\pi}\sigma}$

$e(n-l) = d(n-l) - \mathbf{w}_n^H \varphi_{n-l}$ as *posteriori errors* for each $l : 1 \leq l \leq N$

Santa Clara University

School of Engineering

Quaternion Minimum Error Entropy Algorithm

– The gradient of the cost function *J(n)* can be calculated using GHR calculus based on the following equation:

$$\nabla_{\mathbf{w}_n^*} J(n) = \left(\frac{\partial J(n)}{\partial \mathbf{w}_n}\right)^H$$

$$= \left(\tfrac{1}{4\sigma^2}\right) \times \Bigg[\textstyle\sum_{l=1}^{N} \sum_{t=1}^{N} \exp(g_{l,t}(\mathbf{w}_n))$$

$$\times \big[e(n-l) - e(n-t)\big]\big[\varphi_{n-l}^H - \varphi_{n-t}^H\big] \Bigg]^H$$

Quaternion Minimum Error Entropy Algorithm

– By setting $\mathbf{w_0}$ =0 and replacing *exp(g)* with its kernel equivalent the filter output can be calculated as:

$$\mathbf{w}_n = \zeta \sum_{p=0}^{n-1} \sum_{l=1}^{N} \sum_{t=1}^{N} \Bigg(\Big[\kappa_\sigma \big(e(p-l) - e(p-t)\big)\Big] \times \Big[e(p-l) - e(p-t)\Big]\Big[\varphi_{p-l}^H - \varphi_{p-t}^H\Big] \Bigg)^H$$

$$\text{where } \zeta = \mu\sqrt{2\pi}/16\sigma = \eta\frac{1}{4N^2\sigma^2}$$

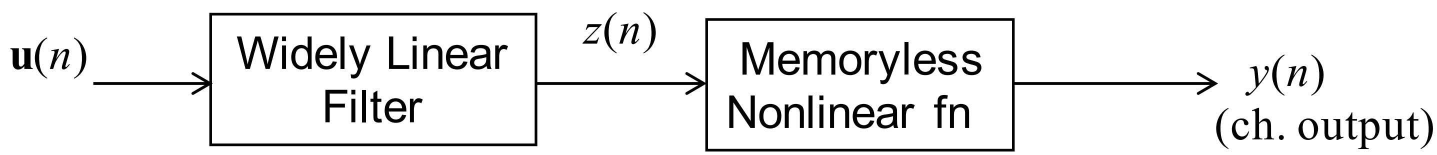
Quaternion Minimum Error Entropy Algorithm

– By substituting the weight update in the $y_n = < \Phi(\mathbf{u}_n)\ , \mathbf{w}_n >= \mathbf{w}_n^H \varphi_n$ 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, we can simplify the equation in kernel form as :

$$y_n = \zeta \sum_{p=0}^{n-1} \sum_{l=1}^{N} \sum_{t=1}^{N} \left[\kappa_\sigma(e(p-l) - e(p-t))\right]$$
$$\times \left[e(p-l) - e(p-t)\right]\left[\bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{p-l}}, \mathbf{u_n}) - \bar{\kappa}_{\bar{\sigma}}(\mathbf{u_{p-t}}, \mathbf{u_n})\right]$$

Simulation

– To evaluate convergence of these algorithms, we used a **channel estimation** task based on the **Weiner nonlinear model**:



– The **widely linear portion** used was:[15]

$$z(n) = g_1^* u(n) + g_2^* u^i(n) + g_3^* u^j(n) + g_4^* u^k(n)$$
$$+ h_1^* u(n-1) + h_2^* u^i(n-1) + h_3^* u^j(n-1) + h_4^* u^k(n-1)$$

– And the **nonlinearity** was:

$$y(n) = z(n) + a z^2(n) + b z^3(n) + v(n)$$

Coeff's:

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

– Widely linear model was used to account second order statistics in quaternion data.

[15] Thomas K. Paul and Tokunbo Ogunfunmi, "A Kernel Adaptive Algorithm for Quaternion-Valued Inputs," in Transactions on Neural Networks and Learning Systems.IEEE, 2015, Volume: 26, Issue: 10, pp. 2422 - 2439 .

Simulation

- 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 (RV) with components from different real Gaussian distributions was formed. The probability distributions used were:

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

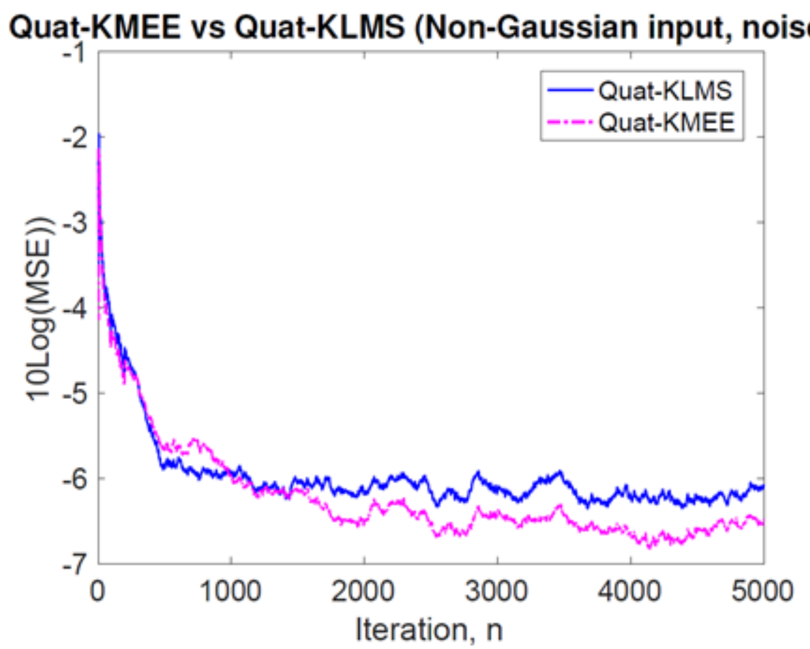
$$
\begin{aligned}
p_v(i) &= (0.90N(0.0, 0.01) + 0.10N(1.0, 0.01)) \\
&+ i(0.70N(3.0, 0.01) + 0.30N(0.5, 0.01)) \\
&+ j(0.45N(1.0, 0.01) + 0.55N(4.5, 0.01)) \\
&+ k(0.80N(0.5, 0.01) + 0.20N(1.5, 0.01))
\end{aligned}
$$

- In simulation non-Gaussian signals were used to show that the Minimum Error Entropy criterion generates more concentrated error peaks in error PDF whereas the variance (MSE) generates wider error distributions.

[8] Felipe A. Tobar and Danilo P. Mandic, "Quaternion Reproducing Kernel Hilbert Spaces: Existence and Uniqueness Conditions," in Transactions on Information Theory . IEEE,2014, Volume: 60, Issue: 9, pp. 5736 - 5749.

Simulation

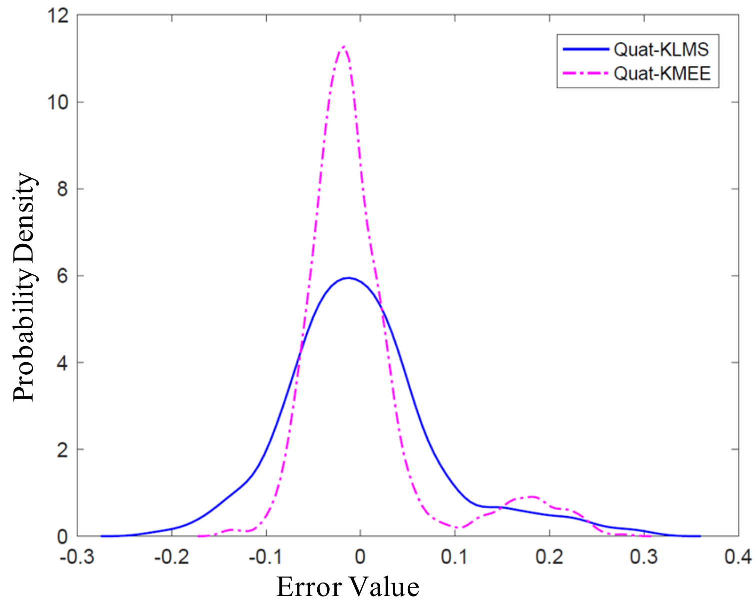
- The results show improvement of Quat-KMEE for modeling nonlinear channel when the input noise is non-Gaussian compared with Quat-KLMS.
- The parameter for the Quat-KMEE were η =0.35, $\bar{\sigma}$ =2.24, σ =0.736
- The Parameter for the Quat-KLMS were η =0.35, $\bar{\sigma}$ =2.24,



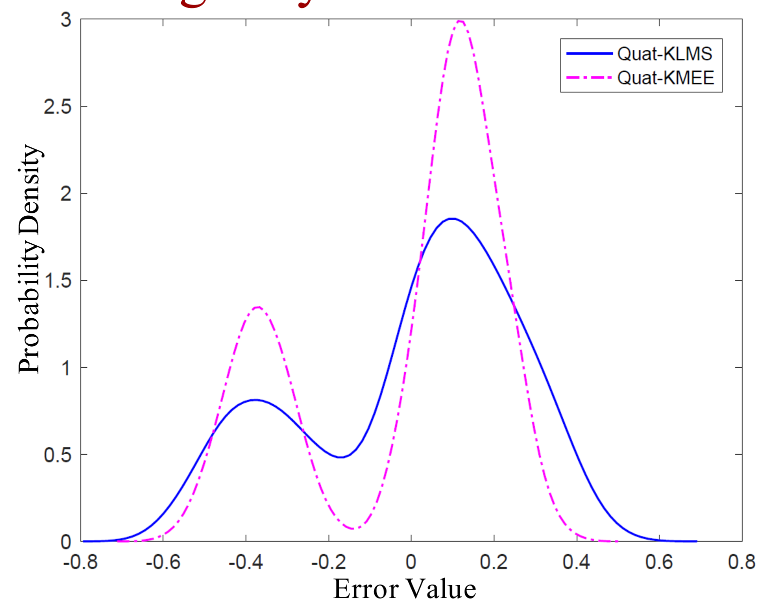
Santa Clara University
School of Engineering

Errors Probability distribution functions of Quat-KLMS and Quat-KMEE for non-Gaussian signal

Real side of errors



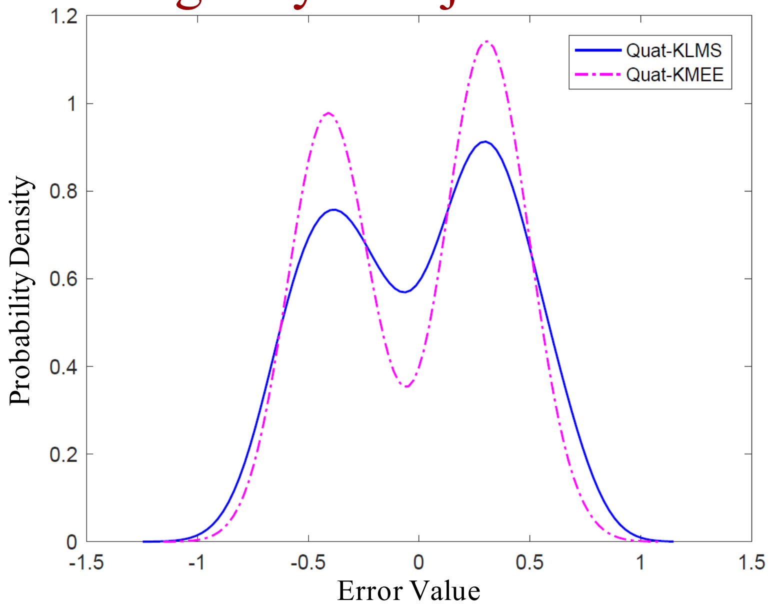
Imaginary side i of errors



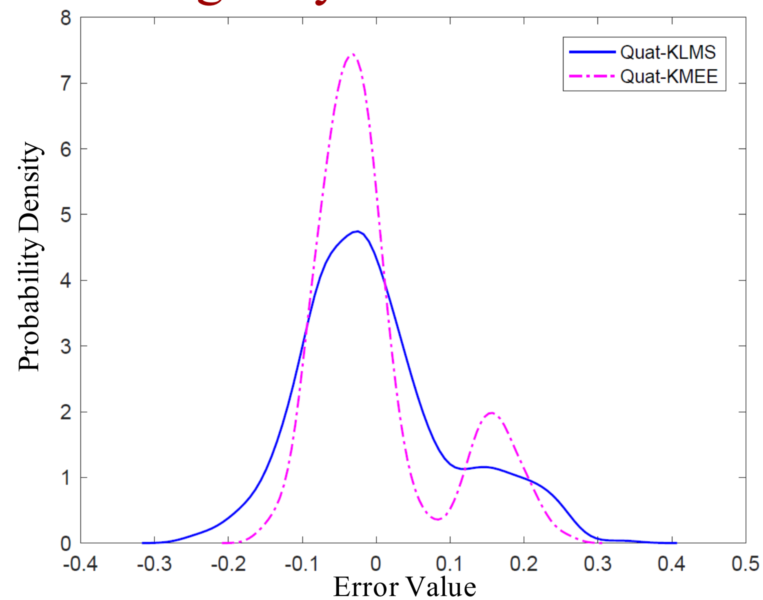
Minimum Error Entropy criterion generates more concentrated error peaks whereas the variance (MSE) generates wider error distributions.

Errors Probability distribution functions of Quat-KLMS and Quat-KMEE for non-Gaussian signal

Imaginary side j of errors



Imaginary side k of errors



Minimum Error Entropy criterion generates more concentrated error peaks whereas the variance (MSE) generates wider error distributions.

Conclusions

- Derivation and demonstration of convergence of a quaternion kernel adaptive algorithm based on minimum error entropy Quat-KMEE based on information theoretic learning (ITL).
- Using GHR calculus for Gradients based on quaternion RKHS.
- Simulation results show the convergence curve of the mean square error of the new algorithm (QKMEE) versus the existing algorithm (QKLMS) that QKMEE has similar speed but better misadjustment.
- The QKMEE algorithm performed better with non-Gaussian signals compared to QKLMS which is based on the MSE criteria adaptive filter and the entropy generates more concentrated error peaks, whereas the variance (MSE) generates wider error distributions.

Thank You