Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

# The q-Least Mean Squares algorithm

U.M. Al-Saggaf [a,b], M. Moinuddin [a,b], M. Arif [c], A. Zerguine [d]

[a] Electrical and Computer Engineering Department, King Abdulaziz University, Saudi Arabia
[b] Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdulaziz University, Saudi Arabia
[c] Electrical Engineering Department, PAF Karachi Institute of Economics and Technology University, Pakistan
[d] Electrical Engineering Department, King Fahd University of Petroleum & Minerals, Saudi Arabia

ARTICLE INFO

*Article history:*
Received 3 May 2014
Received in revised form
8 September 2014
Accepted 24 November 2014

*Keywords:*
Adaptive filters
*q*-gradient
*q*-LMS algorithm
LMS algorithm
Transient analysis
Steady-state analysis

ABSTRACT

The Least Mean Square (LMS) algorithm inherits slow convergence due to its dependency on the eigenvalue spread of the input correlation matrix. In this work, we resolve this problem by developing a novel variant of the LMS algorithms based on the *q*-derivative concept. The *q*-gradient is an extension of the classical gradient vector based on the concept of Jackson's derivative. Here, we propose to minimize the LMS cost function by employing the concept of *q*-derivative instead of the convent ional derivative. Thanks to the fact that the *q*-derivative takes larger steps in the search direction as it evaluates the secant of the cost function rather than the tangent (as in the case of a conventional derivative), we show that the *q*-derivative gives faster convergence for $q>1$ when compared to the conventional derivative. Then, we present a thorough investigation of the convergence behavior of the proposed *q*-LMS algorithm and carry out different analyses to assess its performance. Consequently, new explicit closed-form expressions for the mean-square-error (MSE) behavior are derived. Simulation results are presented to corroborate our theoretical findings.





## 1. Introduction

The concept of adaptive filtering constitutes an important part in statistical signal processing. Whenever there is a requirement to process signals that result from unknown statistics of an environment, the use of an adaptive filter offers an attractive solution to the problem. Thus, adaptive filters are successfully applied in such diverse fields as equalization, noise cancelation, linear prediction, and in system identification [1,2]. The most widely used algorithm for adaptive filters is the Least Mean Squares (LMS) algorithm [3]. The conventional LMS algorithm is derived using the concept of the steepest descent approach with the aid of conventional gradient[1] whose weight update can be formulated as [1]

$$\mathbf{w}_{i+1}=\mathbf{w}_i-\frac{\mu}{2}\nabla_{\mathbf{w}}J(\mathbf{w}), \tag{1}$$

where $J(\mathbf{w})=E[e_i^2]$ for the well known LMS algorithm [1,2] and $e_i$ is the estimation error between the desired response, $d_i$, and its estimate, $\mathbf{u}_i^T\mathbf{w}_i$, produced by an adaptive filter for an input $\mathbf{u}_i$ at time instant $i$, that is,

$$e_i=d_i-\mathbf{u}_i^T\mathbf{w}_i. \tag{2}$$

Since the LMS algorithm belongs to the class of stochastic gradient type adaptive algorithms, it inherits their low computational complexity and their slow convergence, especially

*E-mail addresses:* usaggaf@kau.edu.sa (U.M. Al-Saggaf), mmsansari@kau.edu.sa (M. Moinuddin), mstel11@yahoo.com (M. Arif), azzedine@kfupm.edu.sa (A. Zerguine).

[1] For a function $f(\mathbf{x})$ of a real valued vector $\mathbf{x}=[x_1,\ldots,x_M]^T$, the gradient is defined as $\nabla_{\mathbf{x}}f(\mathbf{x})\triangleq[\partial f/\partial x_1,\ldots,\partial f/\partial x_M]^T$.

http://dx.doi.org/10.1016/j.sigpro.2014.11.016
0165-1684/© 2014 Published by Elsevier B.V.

when operating on highly correlated signals like speech. One approach to overcome the slow convergence problem of the LMS algorithm is by employing a time varying step size in the standard LMS algorithm [4–9]. This is based on using a large step size when the algorithm is far from the optimal solution, thus speeding up the convergence rate, and when the algorithm is near the optimum, a small step size is used to achieve a low level of misadjustment, thus achieving a better overall performance. This can be obtained by adjusting the step size in accordance to some criterion. Several criteria have been used, such as squared instantaneous error [4], sign changes of successive samples of the gradient [5], cross correlation of input and error [6], gradient of squared error cost function [7], and square of the time averaged estimate of the correlation of the error [8], just to name a few. The second approach to improve the convergence speed is to use a normalization in the weight update of the LMS or the Least Mean Fourth (LMF) algorithms, such as used in the normalized LMS (NLMS) algorithm [10] and in the variable XE-NLMF algorithm [11]. Unlike the previous two approaches, a third approach relies on adding a proper constraint to the cost function of the LMS or LMF algorithms [12–15]. Or, more recently, the kernel-based non-linear kernel LMS variants such as the Kernel LMS algorithm for real-valued input [16], the Complex Kernel LMS (CKLMS) algorithm [17] and a modified CKLMS based on modified Wirtinger's Calculus [18] have also been investigated. All these variants of the LMS algorithm improve convergence speed and/or reduce the mean-square-error at the expense of an increase in the computational complexity. In order to improve more the convergence performance of the conventional LMS algorithm while retaining its simplicity, here we propose to utilize a novel concept based on the $q$-calculus which is introduced in the ensuing section, and eventually yield the $q$-LMS algorithm.

### 1.1. Overview of the q-calculus and the q-gradient

In the last few decades, the $q$-calculus has gained a lot of interest in various fields of science, mathematics, physics, quantum theory, statistical mechanics, and signal processing [19]. Jackson introduced the concepts of the $q$-derivative [20] (well known as Jackson's derivative) and the $q$-integral [21]. The $q$-derivative of a function $f(x)$ with respect to variable $x$, denoted by $D_{q,x}f(x)$, is defined as [22]

$$D_{q,x}f(x) \triangleq \begin{cases} \frac{f(qx)-f(x)}{qx-x} & \text{if } x \neq 0, \\ \frac{df(0)}{dx}, & x=0, \end{cases} \tag{3}$$

where $q$ is a real positive number different from 1. In the limiting case of $q \to 1$, the $q$-derivative reduces to the classical derivative. Thus, as an example, the $q$-derivative of a function of the form $x^n$ is

$$D_{q,x}x^n = \begin{cases} \frac{q^n-1}{q-1}x^{n-1} & \text{if } q \neq 1, \\ nx^{n-1} & \text{if } q = 1. \end{cases} \tag{4}$$

Extending this idea to the $q$-gradient of a function $f(\mathbf{x})$ of $n$ variables, where $\mathbf{x}=[x_1,x_2,\ldots,x_n]^T$, the $q$-gradient in this case is defined as

$$\nabla_{\mathbf{q},\mathbf{x}}f(\mathbf{x}) \triangleq [D_{q_1,x_1}f(\mathbf{x}), D_{q_2,x_2}f(\mathbf{x}), \ldots, D_{q_n,x_n}f(\mathbf{x})]^T, \quad \text{for } q \neq 1, \tag{5}$$

where $\mathbf{q}=[q_1,q_2,\ldots,q_n]^T$.

Using the concept of $q$-gradient, it is shown in [23] that the use of the negative of the $q$-gradient of the objective function as the search direction for unconstrained global optimization gives better results than the one obtained by the conventional gradient. This motivates us to investigate the $q$-gradient-based adaptive algorithms.

### 1.2. Paper contributions and organization

The main contributions of the paper are as follows:

(1) In this work, we introduce a new class of adaptive filtering based on $q$-calculus. More specifically, we derive a novel variant of the LMS algorithm by replacing the conventional gradient in (1) by the $q$-gradient which we named as $q$-LMS algorithm.
(2) We provide a geometrical interpretation of the $q$-gradient to justify the proposed design. This also offers us a better understanding that how the $q$-gradient can improve the convergence speed of an adaptive filter.
(3) We show an interesting attribute of the $q$-gradient based LMS algorithm that it can whiten the colored input of the adaptive filter by employing proper selection of its $q$-parameters. Consequently, it improves the convergence speed of the algorithm.
(4) We carry out a thorough analytical investigation of the proposed algorithm by studying both its transient and steady-state convergence behaviors. Consequently, both the MSE and MSD learning curves are evaluated and expressions for the steady-state EMSE and the MSD are derived.
(5) We also develop an efficient mechanism to make the $q$ parameter time varying such that variable $q$-LMS algorithm should give a faster convergence while attaining a lower steady-state EMSE.
(6) We perform extensive simulations to show the superiority of the $q$-LMS algorithms over the conventional LMS and the NLMS algorithms and to validate the analytical results.

The paper is organized as follows. Following this introduction, the $q$-steepest descent algorithm is developed in Section 2. A geometrical interpretation of the $q$-gradient is presented in Section 3. Section 4 introduces the proposed $q$-LMS algorithm. In Section 5, whitening property of the $q$-LMS algorithm is investigated. A thorough performance analysis is carried out for the developed $q$-LMS algorithm in Section 6. In Section 7, an efficient time varying $q$-LMS algorithm is designed. While the simulation results are presented in Section 8, Section 9 summarizes this work.



Please cite this article as: U.M. Al-Saggaf, et al., The q-Least Mean Squares algorithm, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.11.016

## 2. The *q*-steepest descent algorithm

In this section, we design a new class of steepest descent algorithm by replacing the conventional gradient in (1) with the $q$-gradient and calling it $q$-steepest descent algorithm. To set up the stage for derivation, consider a system identification scenario in which the desired response $d_i$ is generated as

$$d_i = \mathbf{u}_i^T \mathbf{w}_o + \eta_i, \tag{6}$$

where $\eta_i$ is a zero mean i.i.d. noise sequence with variance $\sigma_\eta^2$ and $\mathbf{w}_o$ is the unknown system to be identified. Given a sequence of desired response $\{d_i\}$ and a sequence of input regressor vectors $\{\mathbf{u}_i\}$, an adaptive filter generates a weight vector $\mathbf{w}_i$ at each instant so that $\mathbf{u}_i^T \mathbf{w}_i$ is a good estimate of $d_i$ by minimizing the cost function $J(\mathbf{w}) = E[e_i^2]$. To design the weight update of an adaptive filter according to Steepest Descent criteria, we replace the conventional gradient by the $q$-gradient in (1), that is,

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \frac{\mu}{2} \nabla_{\mathbf{q},\mathbf{w}} J(\mathbf{w}). \tag{7}$$

Now, by employing the $q$-gradient's definition provided in (5) with the aid of $q$-derivative rule given in (4), the $\nabla_{\mathbf{q},\mathbf{w}} J(\mathbf{w})$ is evaluated to be

$$\nabla_{\mathbf{q},\mathbf{w}} J(\mathbf{w}) = -2E[\mathbf{G}\mathbf{u}_i e_i], \tag{8}$$

where $\mathbf{G}$ is a diagonal matrix whose $l$th diagonal entry is $g_l = (q_l+1)/2$, that is,

$$\begin{aligned} \text{diag}(\mathbf{G}) &= [g_1, g_2, \ldots, g_M]^T \\ &= \left[\frac{(q_1+1)}{2}, \frac{(q_2+1)}{2}, \ldots, \frac{(q_M+1)}{2}\right]^T. \end{aligned} \tag{9}$$

Substituting the value of $e_i$ in (8) results in the weight update rule of $q$-steepest descent algorithm which is governed by

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \mathbf{G}[\mathbf{r}_{du} - \mathbf{R}_u \mathbf{w}_i], \tag{10}$$

where $\mathbf{R}_u$ is the input auto-correlation matrix and $\mathbf{r}_{du}$ is the cross correlation vector between desired response $d_i$ and input vector $\mathbf{u}_i$. The adaptive rule of the $q$-steepest descent algorithm obtained in (10) is analogous to the conventional steepest descent algorithm except the diagonal matrix $\mathbf{G}$. By analyzing (10) we conclude some important observations in the following remarks:

**Remarks.**

(1) Since the $q$-derivative reduces to the conventional derivative for $q=1$, the $q$-steepest descent algorithm defined in (10) also reduces to the conventional steepest descent algorithm with $q_l = 1$ for all $l = 1, 2, \ldots, M$.

(2) It is shown in Appendix A that the optimum solution for the $q$-steepest descent algorithm results is identical to the solution provided by the conventional steepest descent algorithm, that is, $\mathbf{w}_o = \mathbf{R}_u^{-1} \mathbf{r}_{du}$. Thus, the $q$-steepest descent algorithm promises to attain the same optimum solution as given by the well known Wiener–Hopf equation.

(3) Comparing the $q$-steepest descent algorithm in (10) with its conventional counterpart, it can be noticed that the $q$-steepest descent algorithm has an extra multiplying matrix $\mathbf{G}$. In order to see how this can enhance the convergence speed of the algorithm, in the ensuing section we provide some inference from a geometrical interpretation of the $q$-gradient-based adaptive algorithm.

## 3. Geometrical interpretation of the *q*-gradient based adaptive filtering

To see how the $q$-gradient is beneficial for improving the convergence speed of an adaptive algorithm, we investigate the transient change in the $q$-gradient over the error surface of its cost function $J(\mathbf{w}) = E[e_i^2]$. To set up the stage, we formulate the cost function in terms of the weight vector $\mathbf{w}$ by substituting the expressions of $e_i$ and $d_i$ from (2) and (6), respectively, which results in

$$J(\mathbf{w}_i) = J_{\min} + (\mathbf{w}_i - \mathbf{w}_o)^T \mathbf{R}_u (\mathbf{w}_i - \mathbf{w}_o), \tag{11}$$

where $J_{\min} = \sigma_\eta^2$. Now, consider the scenario of a single tap filter, that is, $\mathbf{w}_i = w_i$, $\mathbf{w}_o = w_o$, and $\mathbf{R}_u = \lambda$. Thus, the above expression for the cost function can be set up as

$$J(w_i) = \sigma_\eta^2 + (w_i - w_o)^2 \lambda, \tag{12}$$

which is a quadratic function in $w_i$. Similarly, the $q$-gradient in (8) can be simplified using the Wiener solution ($\mathbf{w}_o = \mathbf{R}_u^{-1} \mathbf{r}_{du}$) for this single tap filter as

$$\nabla_{q,w} J(w) = -(q+1)\lambda(w_o - w_i). \tag{13}$$

Next, to investigate the behavior of the above gradient along for the cost function (12), we assume that $w_o = 1.5$ and $\sigma_\eta^2 = 0.01$ and we initialize $w_i$ with 0.01. We first simulate the cost function given in (12) by varying $w_i$ in the range [0, 3] as shown in Fig. 1. It can be observed that the cost function is convex and has a parabolic shape (as expected from (12)). Then, we plot the $q$-gradient given in (13) for three values of $q$ which are 10, 3 and 1 (which corresponds to the conventional gradient) for three iterations (i.e., $i = 1, 2, 3$). As depicted from Fig. 1, the $q$-gradient for $q=10$ (black colored triangles) gives a larger change when compared to the ones obtained by $q=3$ (red colored star) and $q=1$ (green colored circle). It can be noticed from the definition of $q$-derivative in (3) that this definition is

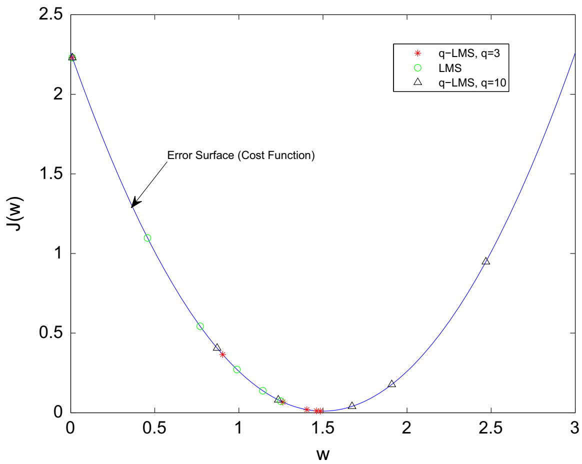


**Fig. 1.** Geometrical interpretation of the $q$-gradient. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

the expression for a secant when $q \neq 1$ and it reduces to a tangent for $q=1$. Knowing the fact that a tangent evaluates the rate of change of a function at a single point when compared to a secant which evaluates the slope of the line joining two points, we can easily infer that the tangent gives a smaller change to the function value compared to the one obtained via a secant. Thus, the $q$-gradient for $q=10$ takes larger steps when compared to the ones obtained by $q=3$ and $q=1$.

## 4. The *q*-Least Mean Squares algorithm

In this section, we derive the $q$-Least Mean Squares ($q$-LMS) algorithm. Dropping the expectation from the $q$-gradient in (8) and using its instantaneous value will result in

$$\nabla_{\mathbf{q},\mathbf{w}} J(\mathbf{w}) \approx -2\mathbf{G}\mathbf{u}_i e_i. \tag{14}$$

Consequently, substituting (14) in (7) results in the $q$-LMS algorithm:

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \mathbf{G}\mathbf{u}_i e_i. \tag{15}$$

For the sake of completeness, by contrasting the above with the standard LMS algorithm, it can be deduced that the $q$-LMS algorithm has an extra degree of freedom to control its performance via the diagonal matrix $\mathbf{G}$ which comprises $q$-dependent entries (see (9) for its definition). Ultimately, the weight update rule in (15) can be set up as

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \overline{\mathbf{u}}_i e_i, \tag{16}$$

where $\overline{\mathbf{u}}_i = \mathbf{G}\mathbf{u}_i$.

Now, observing (16), the $q$-LMS algorithm can be alternately treated as the LMS algorithm with a transformed input vector $\overline{\mathbf{u}}_i = [u_i(1)(q_1+1)/2, u_i(2)(q_2+1)/2, \ldots, u_i(M)(q_M+1)/2]^T$. Hence, the role of the $q$ parameters in $\overline{\mathbf{u}}_i$ can be thought as to transform the given input vector in such a direction that can enhance the performance of the proposed algorithm. For example, one interesting feature of the $q$-LMS algorithm is to increase the convergence speed by a proper selection of the $q$ parameters. Another example is the whitening process, that is, how the $q$-LMS algorithm can be used to whiten a colored input. This issue is discussed next.

### 4.1. Computational complexity of the q-LMS algorithm

One of the important parameters to contrast the performance of an adaptive algorithms is their computational complexity. Since we are employing the $q$ derivative to improve the performance of the conventional LMS algorithm, we compared the computational complexity of the proposed $q$-LMS algorithm with that of the conventional LMS algorithm. To do so, we first reformulate the weight update rule of the $q$-LMS algorithm:

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \mathbf{v} \odot \mathbf{u}_i e_i. \tag{17}$$

where $\odot$ represents the element by element multiplication. Hence, at this stage we can easily contrast the computational complexity of the two algorithms. Specifically, for the real valued data, the conventional LMS algorithm requires $2M+1$ real multiplications and $2M$ real additions per iteration whereas the $q$-LMS algorithm needs $3M+1$ real multiplications and $2M$ real additions per iteration. Thus, the $q$-LMS algorithm requires only $M$ number of multiplications more than the conventional LMS per iteration which does not increase the complexity much as compared to the performance improvement.

## 5. The *q*-gradient based LMS algorithm as a whitening filter

It is well known that the conventional LMS algorithm depends on the input correlation matrix, and therefore its convergence speed is limited by the eigenvalue spread[2] of the input correlation matrix. More specifically, the overall time constant,[3] $\tau_a$, of a mean weight error tap ($v_l(i)$) of the LMS algorithm is bounded by [2]

$$\frac{-1}{\ln(1-\mu\lambda_{\max})} \le \tau_a \le \frac{1}{\ln(1-\mu\lambda_{\min})} \tag{18}$$

where ln( ) represents the natural logarithm, and $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of the input correlation matrix, respectively.

Motivated by the above observation, we found an interesting application of the $q$-gradient. Specifically, we can increase the convergence speed of the LMS algorithm, by selecting the $q$ parameter in such a way that makes the LMS filter acts as a whitening filter. To see this effect, the transient behavior of the $l$th element of the weight error vector of the $q$-LMS given in (58) is investigated. It can be observed that the time constant associated with the $l$th mean weight error tap ($v_l(i)$) is given by

$$\tau_l = \frac{-1}{\ln\left(1 - \dfrac{\mu(q_l+1)\lambda_l}{2}\right)}, \quad 1 \le l \le M. \tag{19}$$

Thus, if we select the $q_l$ parameter such that

$$\frac{(q_l+1)}{2} = \frac{1}{\lambda_l}, \quad \text{or} \quad q_l = \frac{2-\lambda_l}{\lambda_l}, \quad 1 \le l \le M, \tag{20}$$

which modifies the time constant $\tau_l$ to

$$\tau_l = \frac{-1}{\ln(1-\mu)}, \quad 1 \le l \le M, \tag{21}$$

then, the time constant of tap weight becomes independent of the input correlation matrix, and hence this will remove the restriction on the overall time constant defined in (18). Eventually, this will increase the convergence speed of the proposed adaptive algorithm. In other words, this will have a similar effect as the normalized LMS (NLMS) algorithm [10] had on the LMS algorithm. Now, with this choice of the $q$ parameter, the condition for mean stability can be shown to be governed by

$$0 < \mu < 2, \tag{22}$$

[2] Eigenvalue spread is the ratio of maximum eigenvalue to the minimum eigenvalue of the correlation matrix, i.e., eigenvalue spread $= \lambda_{\max}/\lambda_{\min}$.

[3] The time constant $\tau_l$ of a mean weight error tap $v_l(i)$ defines the number of iterations required for its magnitude to reduce by $1/e$ of its initial value $v_l(0)$.



Please cite this article as: U.M. Al-Saggaf, et al., The $q$-Least Mean Squares algorithm, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.11.016

which is identical to that of the NLMS algorithm [10].

## 6. Performance analyses of the *q*-LMS algorithm

In this section, we carry out the mean and mean-square performance analyses of the $q$-LMS algorithm by defining the weight error vector as $\tilde{\mathbf{w}}_i = \mathbf{w}_o - \mathbf{w}_i$ which allows us to set up the weight error recursion for the $q$-LMS algorithm given in (15) as

$$\tilde{\mathbf{w}}_{i+1} = \tilde{\mathbf{w}}_i - \mu \mathbf{G}\mathbf{u}_i e_i. \tag{23}$$

Next, by using the expression for the desired response given in (6), we can rewrite the expression for $e_i$ in (2) as

$$e_i = \mathbf{u}_i^T \tilde{\mathbf{w}}_i + \eta_i. \tag{24}$$

To proceed for the mean and mean square analyses of the weight error vector of the $q$-LMS algorithm, we set up the stage by putting the following assumptions in order:

**A1** The noise $\eta_i$ is zero mean Gaussian with zero odd moments and with variance $\sigma_\eta^2$. Also, the noise $\eta_i$ is independent of the input signal $\mathbf{u}_i$.
**A2** The sequence of vectors $\mathbf{u}_i$ is i.i.d.
**A3** For the sake of mean-square analysis, we assume the autocorrelation matrix of the input regressor $\mathbf{u}_i$ to be diagonal, that is, $\mathbf{R}_u = \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_M)$.

The above assumptions are well known in the literature and are commonly used [1,2]. In the ensuing, the above assumptions are used to evaluate the mean and mean-square performance of the $q$-LMS algorithm.

### 6.1. Mean behavior

After substituting the value of $e_i$ defined by (24) in (23), we can reformulate (23) as

$$\tilde{\mathbf{w}}_{i+1} = \tilde{\mathbf{w}}_i - \mu \mathbf{G}\mathbf{u}_i\left(\mathbf{u}_i^T \tilde{\mathbf{w}}_i + \eta_i\right). \tag{25}$$

Next taking the expectation of both sides of (25), under **A1** and **A2**, the mean value of the weight-error vector the $q$-LMS algorithm can be shown to be governed by the following recursion:

$$E\left[\tilde{\mathbf{w}}_{i+1}\right] = \left(\mathbf{I} - \frac{\mu}{2}\mathbf{A}\right) E[\tilde{\mathbf{w}}_i], \tag{26}$$

where matrix $\mathbf{A}$ is defined as

$$\begin{aligned} \mathbf{A} &= 2\mathbf{G}E\left[\mathbf{u}_i\mathbf{u}_i^T\right] \\ &= 2\mathbf{G}\mathbf{R}_u. \end{aligned} \tag{27}$$

Thus, the weight error vector converges in the mean provided that the step-size $\mu$ of the $q$-LMS algorithm must satisfy the bound given in (28) which after rearranging the terms can be set up as (the transient behavior of $E[\tilde{\mathbf{w}}_i]$ is detailed in Appendix B)

$$0 < \mu < \frac{4}{\max\{(q_1+1)\lambda_1, \ldots, (q_M+1)\lambda_M\}}. \tag{28}$$

In the case when all $q_i$'s are equal (say equal to $q$), then (28) reduces to

$$0 < \mu < \frac{4}{(q+1)\lambda_{\max}}, \tag{29}$$

and then it is very easy to see that (29) collapses to that of the LMS algorithm when $q=1$, that is,

$$0 < \mu < \frac{2}{\lambda_{\max}}. \tag{30}$$

From (28) and (29), it can be seen that the $q$-LMS algorithm is dependent on the energy of the input signal, as was in the case of the LMS algorithm. Therefore, unlike a colored signal, a white input signal would result in a better performance. To remedy this situation, either a normalized version of this algorithm can be employed or an appropriate selection of $q$ parameters can be used. The later solution was already elaborated in Section 5.

### 6.2. Mean square behavior

Here, we are interested in studying the time-evolution and the steady-state values of $E[||\tilde{\mathbf{w}}_i||_\lambda^2]$ and $E[||\tilde{\mathbf{w}}_i||^2]$ of the $q$-LMS algorithm which represent the excess mean-square-error (EMSE) and the mean-square-deviation (MSD) performances of the filter, respectively, whereas their time evolution relate to the learning or the transient behavior of the filter. To derive these performance measures, we set up the stage by defining error measures and the fundamental weighted energy relation for the $q$-LMS algorithm in the ensuing sections.

#### 6.2.1. Error measures and fundamental weighted-energy relation

For some symmetric positive definite weighting matrix $\boldsymbol{\Sigma}$ to be specified later, the weighted a priori and a posteriori estimation errors are, respectively, defined as [1]

$$e_{ai}^{\boldsymbol{\Sigma}} \triangleq \mathbf{u}_i^T \boldsymbol{\Sigma} \tilde{\mathbf{w}}_i, \quad \text{and} \quad e_{pi}^{\boldsymbol{\Sigma}} \triangleq \mathbf{u}_i^T \boldsymbol{\Sigma} \tilde{\mathbf{w}}_{i+1}. \tag{31}$$

For the special case when $\boldsymbol{\Sigma} = \mathbf{I}$ ($\mathbf{I}$ is the identity matrix), the weighted a priori and a posteriori estimation errors defined above are reduced to standard a priori and a posteriori estimation errors, respectively, that is,

$$e_{ai} = e_{ai}^{\mathbf{I}} = \mathbf{u}_i^T \tilde{\mathbf{w}}_i, \quad \text{and} \quad e_{pi} = e_{pi}^{\mathbf{I}} = \mathbf{u}_i^T \tilde{\mathbf{w}}_{i+1}. \tag{32}$$

Observing (24), it can be seen that the estimation error, $e_i$, and the a priori error, $e_{ai}$, are related via $e_i = e_{ai} + \eta_i$. Thus, by employing the opted assumptions, it can be shown that $E[e_i^2] = E[||\tilde{\mathbf{w}}_i||_\lambda^2] + \sigma_\eta^2$. Thus, the term $E[||\tilde{\mathbf{w}}_i||_\lambda^2]$ gives the EMSE.

To perform the mean-square analysis of the $q$-LMS algorithm, we develop the fundamental weighted-energy relation using the methodology outlined in [1]. As a result, the fundamental weighted-energy relation for $q$-LMS algorithm is found to be (the proof is provided in Appendix C)

$$E[||\tilde{\mathbf{w}}_{i+1}||_{\boldsymbol{\sigma}}^2] = E[||\tilde{\mathbf{w}}_i||_{\mathbf{F}\boldsymbol{\sigma}}^2] + \mu^2 \sigma_\eta^2 \boldsymbol{\lambda}^T \mathbf{G}^2 \boldsymbol{\sigma}, \tag{33}$$

where $\boldsymbol{\lambda} = \mathrm{diag}(\boldsymbol{\Lambda})$ is a column vector containing diagonal entries of $\boldsymbol{\Lambda}$ and $\boldsymbol{\sigma}$ is an $M \times 1$ parameter weight vector that can provide different performance measures by choosing its appropriate value as will be shown in the next section. The matrix $\mathbf{F}$ is given by

$$\mathbf{F} = \mathbf{I} - \mu \mathbf{A} + \mu^2 \mathbf{B}, \tag{34}$$



Please cite this article as: U.M. Al-Saggaf, et al., The *q*-Least Mean Squares algorithm, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.11.016

where

$$\mathbf{B} = 2\mathbf{G}^2\mathbf{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\mathbf{G}^2. \tag{35}$$

#### 6.2.2. Learning curves for the EMSE and the MSD of the q-LMS algorithm

Now, we deduce the EMSE and the MSD learning curves of the proposed algorithm by selecting the proper choice of $\boldsymbol{\sigma}$ defined in (33). Starting with an initial value of weight vector $\mathbf{w}_{-1}$ equal to zero vector (consequently $\tilde{\mathbf{w}}_{-1} = \mathbf{w}^o$), we can obtain the EMSE learning curve by setting $\boldsymbol{\sigma}$ equal to $\boldsymbol{\lambda}$ in (33) and is found to be

$$\begin{aligned}\mathrm{EMSE}(i) &= E[\|\tilde{\mathbf{w}}_i\|^2_{\boldsymbol{\lambda}}]\\ &= E[\|\mathbf{w}^o\|^2_{\mathbf{F}^i\boldsymbol{\lambda}}] + \mu^2\sigma_\eta^2 E\left[\|\mathbf{u}_i\|^2_{(\mathbf{I}+\mathbf{F}+\cdots+\mathbf{F}^{i-1})\mathbf{G}^2\boldsymbol{\lambda}}\right]\\ &= E[\|\mathbf{w}^o\|^2_{\mathbf{F}^i\boldsymbol{\lambda}}] + \mu^2\sigma_\eta^2\boldsymbol{\lambda}^T(\mathbf{I}+\mathbf{F}+\cdots+\mathbf{F}^{i-1})\mathbf{G}^2\boldsymbol{\lambda}.\end{aligned} \tag{36}$$

whereas the MSD learning curve is obtained by setting $\boldsymbol{\sigma}$ equal to $\mathbf{1}$ in (33) and it is given by[4]

$$\begin{aligned}\mathrm{MSD}(i) &= E[\|\tilde{\mathbf{w}}_i\|^2_{\mathbf{1}}]\\ &= E[\|\mathbf{w}^o\|^2_{\mathbf{F}^i\mathbf{1}}] + \mu^2\sigma_\eta^2\boldsymbol{\lambda}^T(\mathbf{I}+\mathbf{F}+\cdots+\mathbf{F}^{i-1})\mathbf{G}^2\mathbf{1}.\end{aligned} \tag{37}$$

#### 6.2.3. Mean square stability

Following the strategy outlined in [1], we can prove that the mean square stability of the q-LMS algorithm is conditioned by the following bound:

$$0 < \mu < \frac{1}{\lambda_{\max}(\mathbf{A}^{-1}\mathbf{B})}, \tag{38}$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined in (27) and (35), respectively.

#### 6.2.4. Steady-state performance

In this section, we evaluate the steady state performance of the proposed algorithm by analyzing (33) as $i \to \infty$ and consequently derive expressions for the steady-state EMSE and MSD. As $i \to \infty$ the terms $E[\|\tilde{\mathbf{w}}_i\|^2_{\boldsymbol{\sigma}}]$ and $E[\|\tilde{\mathbf{w}}_{i-1}\|^2_{\mathbf{F}\boldsymbol{\sigma}}]$ can be combined as $E[\|\tilde{\mathbf{w}}_\infty\|^2_{(\mathbf{I}-\mathbf{F})\boldsymbol{\sigma}}]$ to obtain

$$E[\|\tilde{\mathbf{w}}_\infty\|^2_{(\mathbf{I}-\mathbf{F})\boldsymbol{\sigma}}] = \mu^2\sigma_\eta^2\boldsymbol{\lambda}^T\mathbf{G}^2\boldsymbol{\sigma}. \tag{39}$$

When $\boldsymbol{\sigma} = (\mathbf{I}-\mathbf{F})^{-1}\boldsymbol{\lambda}$, the steady state EMSE for the above equation is derived to look like

$$\mathrm{EMSE} = \mu^2\sigma_\eta^2\boldsymbol{\lambda}^T\mathbf{G}^2(\mathbf{I}-\mathbf{F})^{-1}\boldsymbol{\lambda}. \tag{40}$$

Similarly, setting $\boldsymbol{\sigma} = \mathbf{1}$ in (39), the steady-state MSD of the proposed algorithm is found to be

$$\mathrm{MSD} = \mu^2\sigma_\eta^2\boldsymbol{\lambda}^T\mathbf{G}^2(\mathbf{I}-\mathbf{F})^{-1}\mathbf{1}. \tag{41}$$

To get more insight, the matrix inversion lemma [2] for the term $(\mathbf{I}-\mathbf{F})^{-1}$ is used to obtain

$$\begin{aligned}(\mathbf{I}-\mathbf{F})^{-1} &= \left(2\mu\mathbf{\Lambda}\mathbf{G} - 2\mu^2\mathbf{G}^2\mathbf{\Lambda}^2 + \mu^2\boldsymbol{\lambda}\boldsymbol{\lambda}^T\mathbf{G}^2\right)^{-1}\\ &= \left(2\mu\mathbf{\Lambda}\mathbf{G}^{-1} - 2\mu^2\mathbf{\Lambda}^2 + \mu^2\boldsymbol{\lambda}\boldsymbol{\lambda}^T\right)^{-1}\mathbf{G}^{-2}\\ &= \frac{(2\mu\mathbf{\Lambda}\mathbf{G}^{-1} - 2\mu^2\mathbf{\Lambda}^2)^{-1}\mathbf{G}^{-2}}{1-\mu^2\boldsymbol{\lambda}^T(2\mu\mathbf{\Lambda}\mathbf{G}^{-1} - 2\mu^2\mathbf{\Lambda}^2)^{-1}\boldsymbol{\lambda}}.\end{aligned} \tag{42}$$

As a result, the EMSE and the MSD simplify, respectively, to

$$\begin{aligned}\mathrm{EMSE} &= \mu^2\sigma_\eta^2\frac{\boldsymbol{\lambda}^T\mathbf{G}^2(2\mu\mathbf{\Lambda}\mathbf{G}^{-1} - 2\mu^2\mathbf{\Lambda}^2)^{-1}\mathbf{G}^{-2}\boldsymbol{\lambda}}{1-\mu^2\boldsymbol{\lambda}^T(2\mu\mathbf{\Lambda}\mathbf{G}^{-1} - 2\mu^2\mathbf{\Lambda}^2)^{-1}\boldsymbol{\lambda}}\\ &= \frac{\mu\sigma_\eta^2\sum_{l=1}^M\frac{\lambda_l g_l}{2(1-\mu\lambda_l)}}{1-\mu\sum_{l=1}^M\frac{\lambda_l g_l}{2(1-\mu\lambda_l)}},\end{aligned} \tag{43}$$

and

$$\begin{aligned}\mathrm{MSD} &= \mu^2\sigma_\eta^2\frac{\boldsymbol{\lambda}^T\mathbf{G}^2(2\mu\mathbf{\Lambda}\mathbf{G}^{-1} - 2\mu^2\mathbf{\Lambda}^2)^{-1}\mathbf{G}^{-2}\mathbf{1}}{1-\mu^2\boldsymbol{\lambda}^T(2\mu\mathbf{\Lambda}\mathbf{G}^{-1} - 2\mu^2\mathbf{\Lambda}^2)^{-1}\boldsymbol{\lambda}}\\ &= \frac{\mu\sigma_\eta^2\sum_{l=1}^M\frac{g_l}{2(1-\mu\lambda_l)}}{1-\mu\sum_{l=1}^M\frac{\lambda_l g_l}{2(1-\mu\lambda_l)}}.\end{aligned} \tag{44}$$

**Remarks.**

(1) Expressions (43) and (44) result in a more restrictive range for the step size $\mu$ when $\mu < 1/\lambda_l$ for all $1 \le l \le M$, that is

$$0 < \mu < \frac{2}{\sum_{l=1}^M \lambda_l g_l}. \tag{45}$$

Moreover, for the case of $g_l = 1$, $\forall l$ (which corresponds to the standard LMS case), the above range of the step-size simplifies to

$$0 < \mu < \frac{2}{\sum_{l=1}^M \lambda_l}. \tag{46}$$

(2) It can be noticed that one can recover the steady-state EMSE and MSD of the conventional LMS algorithm by substituting $q_l = 1$ for all $1 \le l \le M$ or equivalently by setting $\mathbf{G} = \mathbf{I}$ in (40) and (41), respectively.

## 7. An efficient q-LMS algorithm with time varying q parameter

We have seen in Sections 3 and 4 that how the q-gradient with $q > 1$ improves the convergence speed of the adaptive filter. On the other hand, we notice that the performance of the q-LMS algorithm degrades when $q > 1$ as dictated by the expressions of the EMSE and MSD defined given in (43) and (44), respectively. In other words, the larger the value of the q parameter, the faster the convergence of the algorithm at the expense of a degradation in the steady-state performance. This motivates us to design a q-LMS algorithm with a time varying q parameter such that the q parameter attains initially larger value (that is, greater than 1) and reduces to 1 near steady-state. Eventually, this technique will promise both a faster convergence and a lower steady-state value. A similar approach was carried out in [9]. Thus, we propose the following time varying rule for the q parameter:

$$\psi_{i+1} = \beta\psi_i + \gamma e_i^2, \quad (0 < \beta < 1, \gamma > 0), \tag{47}$$

[4] $\mathbf{1}$ is an $M$-dimensional vector with all entries equal to 1.

with

$$q_{i+1} = \begin{cases} q_{upper} & \text{if } \psi_{i+1} > q_{upper} \\ 1 & \text{if } \psi_{i+1} < 1 \\ \psi_{i+1} & \text{otherwise} \end{cases} \tag{48}$$

where $q_{upper}$ is so chosen to satisfy the stability bound, that is,

$$q_{upper} = \frac{2}{\mu \lambda_{max}}. \tag{49}$$

The above scheme provides an automatic adjustment of $q_i$ according to the estimation of the square of the estimation error. When this estimate is a large value, $q_i$ will approach its upper bound denoted by $q_{upper}$, thus providing fast adaptation while its smaller value will make $q_i$ close to unity for a lower steady-state error. Therefore, promising both a faster convergence and a lower steady-state error for the $q$-LMS algorithm.

**Remark.** By comparing the update rule for the time varying $q$-LMS guven in (47) and (48) with the update rule for the variable step-size of the VSS-LMS algorithm [4], it can be easily deduced that the computational complexity of the time varying $q$-LMS is almost the same as that of the VSS-LMS algorithm except $M$ number of multiplications as mentioned in Section 4.1.

## 8. Simulation results

In this section, the performance analysis of the $q$-LMS algorithm is investigated in a system identification scenario with $\mathbf{w}^o = [0.227, 0.460, 0.688, 0.460, 0.227]^T$. System noise is a zero mean i.i.d. sequence with variance 0.001 which set the SNR to 30 dB. Throughout the simulation, the adaptive filter used has the same length as that of the unknown system. The input to the adaptive filter and unknown system is correlated complex Gaussian input which is generated with correlation matrix with entries $\mathbf{R}(i,j) = \alpha_c^{|i-j|}$ with correlation factor[5] $\alpha_c$ $(0 < \alpha_c < 1)$. The objectives of our simulations are as follows:

(1) To investigate the transient trajectories of the $q$-Gradient based MSE cost function.
(2) To compare the MSE performances of the $q$-LMS and the conventional LMS algorithms.
(3) To validate the derived analytical results for both steady-state and transient analysis.
(4) To investigate the performance of the time varying $q$-LMS algorithm.

The above outlined objectives are presented in the ensuing sub-sections.

[5] The case $\alpha_c = 0$ corresponds to the white case while $\alpha_c = 1$ corresponds to the fully correlated case.

### 8.1. Transient trajectories of the q-Gradient based MSE cost function

In this study, we evaluate the transient trajectories of weight error vector and the transient behavior of the cost function. The cost function $J = E[e_i^2]$ can be formulated using weight error vector as

$$J(i) = J_{\min} + \sum_{l=1}^{M} \lambda_l v_l^2(i), \tag{50}$$

where $J_{\min}$ is the minimum value of $J$ evaluated at $\mathbf{w}_o$ which is equal to the noise variance $\sigma_\eta^2$. Now, by substituting $v_l^2(i)$ from (58) in the above, we obtain

$$J(i) = J_{\min} + \sum_{l=1}^{M} \lambda_l \left(1 - \frac{\mu(q_l+1)\lambda_l}{2}\right)^{2i} v_l^2(0). \tag{51}$$

For the purpose of our study, we consider the filter length equal to 2 (i.e., $M=2$) and white Gaussian noise process. The optimum weight vector $\mathbf{w}_o$ and the initial value of weight error vector $\mathbf{v}_0$ are selected by using the approach described in [2]. Here, we investigate two different scenarios of input correlation which are discussed next.

**Example 1.** In this example, we choose $\sigma_\eta^2 = 0.0965$, the optimum weight vector as $\mathbf{w}_o = [0.1950, -0.95]^T$ and the initial weight error vector $\mathbf{v}_0 = [0.5339, -0.8096]^T$. The eigenvalues of the input correlation matrix used are $[\lambda_1, \lambda_2] = [1.1, 0.9]$ (i.e., the eigenvalue spread$=1.22$) which corresponds to low correlated inputs. Fig. 2 shows the rings.

**Example 2.** In the second example, illustrated by Fig. 3, we choose $\sigma_\eta^2 = 0.0038$, the optimum weight vector as $\mathbf{w}_o = [1.9114, -0.95]^T$, and the initial weight error vector $\mathbf{v}_0 = [-0.6798, -2.0233]^T$. The eigenvalues of the input correlation matrix used are $[\lambda_1, \lambda_2] = [1.957, 0.0198]$ (i.e., the eigenvalue spread$=100$) which corresponds to a highly correlated input.

### 8.2. Sensitivity analysis of the q-LMS algorithm

In this experiment, we analyze the sensitivity of the $q$-LMS algorithm with respect to the parameter $q$. To do so, we choose a system identification problem in which the unknown system to be identified is $\mathbf{w}^o = [0.227, 0.460, 0.688, 0.460, 0.227]^T$. In this context, we compare the MSE learning curves of the $q$-LMS algorithm for different fixed values of $q$ and compared it with the one obtained via the conventional LMS algorithm in Fig. 4. The results are averaged over 500 independent runs. For a fair comparison, we set the equal step-size values which is equal to $\mu_{\text{LMS}} = \mu_{q-\text{LMS}} = 0.08$. For the $q$-LMS algorithm, we investigated four fixed values of $q$ which are $q=0.0001$, $q=1$, $q=2$, and $q=3$. It can be depicted from the figure that the result for the case $q=1$ exactly coincides with that of the conventional LMS algorithm which validates that $q=1$ corresponds to standard LMS case. Moreover, it can be seen from the reported results that the $q$-LMS algorithm exhibits faster convergence and a large steady-state MSE for larger $q$ while slower convergence and smaller steady-state MSE for smaller $q$. This shows that the steady-state MSE of the $q$-LMS algorithm is a monotonically

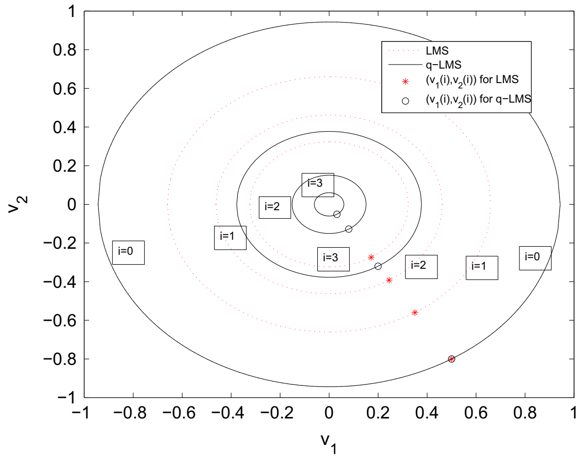


**Fig. 2.** Learning trajectory for the *q*-steepest Descent algorithm with eigenvalue spread = 1.22.

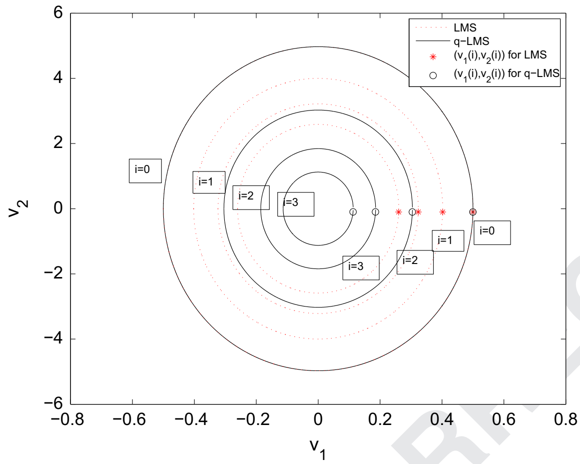


**Fig. 3.** Learning trajectory for the *q*-steepest Descent algorithm with eigenvalue spread = 100.

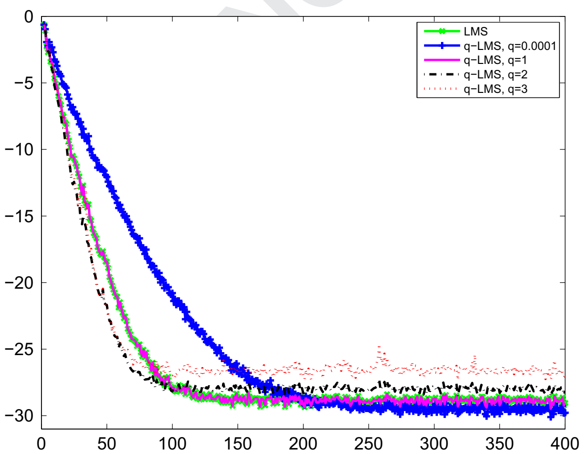


**Fig. 4.** MSE behavior for the fixed q-LMS and the conventional LMS algorithms.

increasing function of $q$. This behavior can be explained with the help of $q$-gradient's concept. As the $q$-gradient computes the secant of a function for a $q$ value greater than 1, which corresponds to taking larger steps towards minima, and therefore results in a faster convergence and vice versa. For the case of $q = 1$, the two MSE learning curves coincide as expected because the $q$-gradient transforms to the ordinary gradient at $q = 1$.

### 8.3. Whitening behavior of the q-LMS algorithm

In this section, we explore the whitening behavior of the $q$-LMS algorithm with a proper selection of the $q$ values. Thus, by setting the $q$ values according to (20) or equivalently by setting $\mathbf{G} = \boldsymbol{\Lambda}^{-1}$, the $q$-LMS algorithm cancels the effect of the input correlation or in other words it whitens the input (see Section 5 for details). In Fig. 5, the MSE learning curve of the $q$-LMS algorithm $\mathbf{G} = \boldsymbol{\Lambda}^{-1}$ is compared to that of the conventional LMS and the NLMS algorithms. The inputs to the adaptive filters are correlated with an eigenvalue spread of 100. It can be easily seen from the results that the $q$-LMS

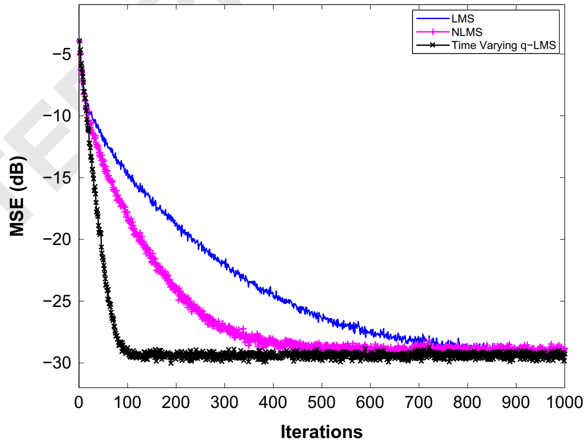


**Fig. 5.** MSE behavior of the whitening *q*-LMS, the conventional LMS and the NLMS algorithms.

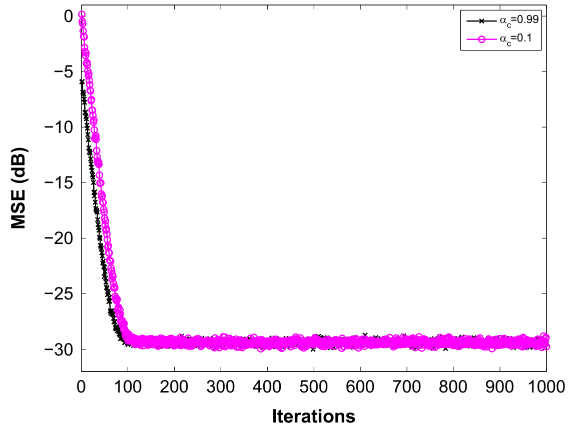


**Fig. 6.** MSE behavior of the whitening *q*-LMS for large and small input correlation.

algorithm outperforms both the NLMS and the conventional LMS algorithms. The reason is that the choice of $q$ parameters, according to (20), makes the convergence of $q$-LMS algorithm completely independent from the input correlation as explained in Section 5. This fact is further supported by the results in Fig. 6. Here, the $q$-LMS algorithm with whitening $q$ selection is simulated for two extreme values of the correlation factor ($\alpha_c$), that is, for $\alpha_c = 0.99$ and $\alpha_c = 0.1$ which correspond to eigenvalue spreads of 885 and 1.41, respectively. There is a clear demonstration in the reported results that the convergence of the $q$-LMS algorithm with the whitening $q$ selection is insensitive to the input correlation.

### 8.4. Validating the derived analytical results for the q-LMS algorithm

In this section, we compared the simulation results with the derived analytical ones in order to validate our theoretical findings. For that, we investigated both the steady-state and the transient performance of the $q$-LMS algorithm. In the first experiment reported in Fig. 7, we compared the simulation MSE learning curves of the $q$-LMS for whitening $q$ with the analytical one obtained from the derived expression in (36) for two values of step-size value which are 0.1 and 0.01. Here, the correlation factor is set to $\alpha_c = 0.5$. An excellent agreement between the theory and simulation can be observed here which validates that our theoretical derivations are valid for both small and large step-size scenarios. In the second experiment shown in Fig. 8, we plotted the analytical values of the steady-state EMSE derived in (40) against the step-size values and compared it with the simulation one for two different choices of matrix $\mathbf{G}$, which are $\mathbf{G} = \mathbf{\Lambda}^{-1}$ (showing whitening $q$-LMS case) and $\mathbf{G} = \mathbf{I}$ (showing the conventional LMS case). Again the results show an excellent match between the theory and the simulations. Moreover, it can be observed that the conventional LMS algorithm has larger steady-state EME'S compared to the whitening $q$-LMS algorithm particularly at larger values of the step-size.

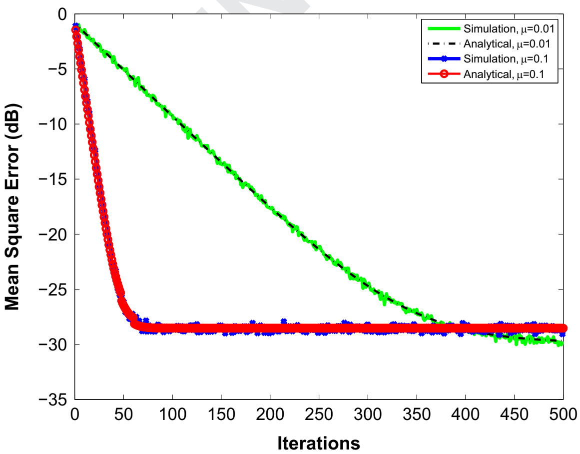


**Fig. 7.** Simulation and analytical MSE behavior of the $q$-LMS algorithm.

### 8.5. Performance of time varying q-LMS algorithm

Finally, we have investigated the performance of the time varying $q$-LMS algorithm developed in Section 7. In order to implement a time varying $q$-LMS algorithm, we use an intelligent mechanism using (47) and (49) which provides an automatic adjustment of $q_i$ according to the energy of estimation error. We set the initial value of $q_i$ according to the whitening criterion, that is, we choose $\mathbf{G}_0 = \mathbf{\Lambda}^{-1}$. The results are compared with that of the conventional LMS algorithm (with $\mu_{LMS} = 0.05$), the NLMS algorithm (with $\mu_{NLMS} = 1$), Variable step-size LMS (VSS-LMS) algorithm [4], and Modified Variable step-size LMS [8]. This comparison is shown in Fig. 9. It can be seen from the figure that the time varying $q$-LMS outperforms the conventional LMS by attaining many fold faster convergence speed. This is due to the fact that by employing the proposed mechanism, $q_i$ attains a larger value in the initial stage of adaptation (due to larger estimation error energy) and it decreases to a smaller value near steady-state (due to small estimation error energy). Hence, it gives a faster convergence in the initial transient stage and a lower

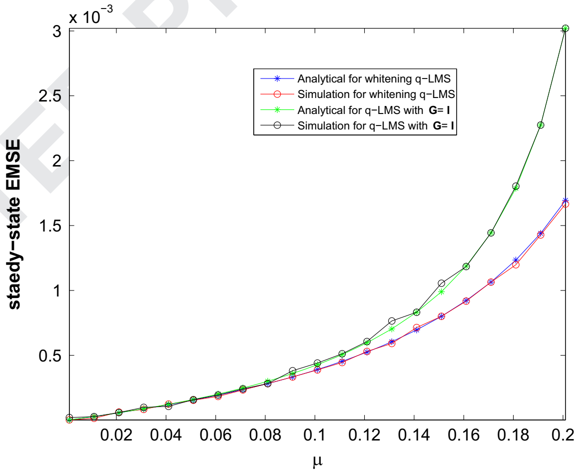


**Fig. 8.** Steady-state EMSE of the $q$-LMS for $\mathbf{G} = \mathbf{\Lambda}^{-1}$ and $\mathbf{G} = \mathbf{I}$.

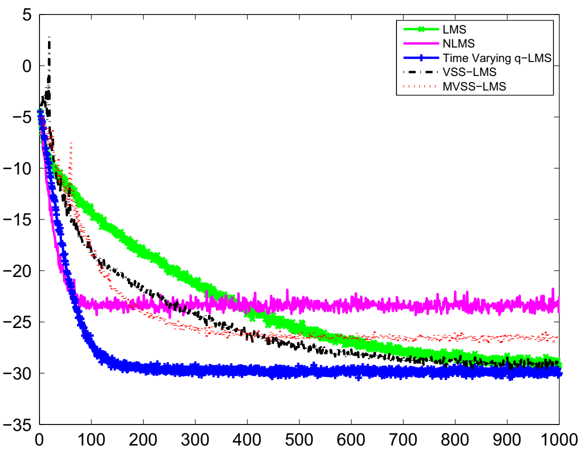


**Fig. 9.** MSE behavior of the time varying $q$-LMS, LMS, NLMS, VSS-LMS, and MVSS-LMS algorithms.



Please cite this article as: U.M. Al-Saggaf, et al., The $q$-Least Mean Squares algorithm, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.11.016

steady-state error near final stages.

## 9. Conclusion

In this work, we developed a novel $q$-LMS algorithm using the concept of $q$-gradient in contrast to the standard gradient in the LMS algorithm. This provides an extra degree of freedom to control the performance of the algorithm in terms of both convergence speed and steady-state error which we proved with the aid of exhaustive simulations. We supported the rationale of the proposed work with the aid of $q$-gradient's geometry. One interesting feature of the proposed algorithm is that it can act like a whitening filter with the proper choice of the $q$-parameters. Mean and MSE performance analyses of the proposed algorithm are also carried out for both the transient and the steady-state scenarios. We also developed a variable $q$-LMS algorithm which gives a faster convergence while attaining a lower steady-state EMSE. We hope that our work has opened a new door in the area of adaptive filtering as a number of existing adaptive algorithms can be investigated in a new paradigm of the $q$-gradient.

## Acknowledgments

This work was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant no. (135-025-D1434). The authors, therefore, acknowledge with thanks DSR technical and financial supports.

## Appendix A. Optimum solution for the $q$-steepest descent algorithm

The optimum solution for the $q$-steepest descent algorithm is derived in this Appendix. Since, the $q$-gradient corresponds to a secant for $q>1$, it becomes a tangent as the value of $q$ approaches unity. This is due to the fact that the $q$-derivative transforms to standard derivative as $q$ becomes unity. Hence, the $q$-gradient promises to attain the minimum value of the cost function as $q\to 1$ which implies that the slope of the tangent approaches zero near the optimum solution. In other words, the $q$-gradient, $\nabla_{\mathbf{q},\mathbf{w}}J(\mathbf{w})$ derived in (8), approaches zero as the $q$ approaches unity, that is,

$$-2E[\mathbf{G}\mathbf{u}_i e_i]\approx 0 \quad \text{as } q\to 1.$$

Upon taking the expectation of the above expression, after substituting the expression for $e_i$, the following is obtained:

$$\mathbf{G}[\mathbf{r}_{du}-\mathbf{R}_u\mathbf{w}_o]\approx 0 \quad \text{as} \quad q\to 1, \tag{52}$$

where $\mathbf{R}_u$ is the input auto-correlation matrix and $\mathbf{r}_{du}$ is the cross correlation vector between the desired response, $d_i$, and the input vector, $\mathbf{u}_i$. Finally, after some simplification steps, the optimum weight vector can be shown to be

$$\mathbf{w}_o\approx \mathbf{R}_u^{-1}\mathbf{r}_{du} \quad \text{as} \quad q\to 1. \tag{53}$$

## Appendix B. Mean weight error vector recursion for $q$-LMS algorithm

In this Appendix, we analyze the transient behavior of mean weight error vector of the $q$-LMS algorithm defined by (26). Now, by defining the variable $\mathbf{v}_i$ as

$$\mathbf{v}_i \triangleq E[\tilde{\mathbf{w}}_i], \tag{54}$$

we can formulate the mean weight error recursion in (26) as

$$\mathbf{v}_{i+1}=\left(\mathbf{I}-\frac{\mu}{2}\mathbf{A}\right)\mathbf{v}_i. \tag{55}$$

Resorting to assumption **A3**, the matrix $\mathbf{A}$ in (26) can be set up as

$$\mathbf{A}=\text{diag}\left(\frac{(q_1+1)\lambda_1}{2},\ldots,\frac{(q_M+1)\lambda_M}{2}\right). \tag{56}$$

Consequently, the $l$th element in the weight error vector $\mathbf{v}_i$ (denoted by $v_l(i)$) will take the following form:

$$v_l(i+1)=\left(1-\frac{\mu(q_l+1)\lambda_l}{2}\right)v_l(i), \quad 1\le l\le M. \tag{57}$$

With an initial value $v_l(0)$, the solution of the above difference equation can be easily shown to be governed by

$$v_l(i)=\left(1-\frac{\mu(q_l+1)\lambda_l}{2}\right)^i v_l(0), \quad 1\le l\le M, \tag{58}$$

which is a geometric series and will converge as $i\to\infty$ provided that

$$\left|1-\frac{\mu(q_l+1)\lambda_l}{2}\right|<1, \quad 1\le l\le M. \tag{59}$$

Thus, the condition for the overall mean convergence of the $q$-LMS algorithm can be obtained when the above bound is satisfied for all the elements in $\mathbf{v}_i$ which is equivalent to say

$$0<\mu\ \max\left\{\frac{(q_1+1)\lambda_1}{2},\frac{(q_2+1)\lambda_2}{2},\ldots,\frac{(q_M+1)\lambda_M}{2}\right\}<2 \tag{60}$$

where the notation max{} represents the maximum quantity among the entries in {}.

## Appendix C. Fundamental weighted variance relation for the $q$-LMS algorithm

In this section, we derive the weighted variance relation. This is done by first developing the fundamental weighted-energy relation. To proceed, we multiply both sides of (23) by $\mathbf{u}_i^T\boldsymbol{\Sigma}\mathbf{G}$ to obtain

$$\mathbf{u}_i^T\boldsymbol{\Sigma}\mathbf{G}\tilde{\mathbf{w}}_{i+1}=\mathbf{u}_i^T\boldsymbol{\Sigma}\mathbf{G}\tilde{\mathbf{w}}_i-\mu e_i\mathbf{u}_i^T\boldsymbol{\Sigma}\mathbf{G}^2\mathbf{u}_i. \tag{61}$$

Now, using the definitions in (31) and replacing $\boldsymbol{\Sigma}$ by $\boldsymbol{\Sigma}\mathbf{G}$, we can rewrite the above equation as[6]

$$e_{pi}^{\boldsymbol{\Sigma}\mathbf{G}}=e_{ai}^{\boldsymbol{\Sigma}\mathbf{G}}-\mu e_i||\mathbf{u}_i||^2_{\boldsymbol{\Sigma}\mathbf{G}^2}. \tag{62}$$

Thus, by substituting Eq. (62) in Eq. (23), we obtain the following relation:

$$\tilde{\mathbf{w}}_{i+1}=\tilde{\mathbf{w}}_i-\frac{(e_{ai}^{\boldsymbol{\Sigma}\mathbf{G}}-e_{pi}^{\boldsymbol{\Sigma}\mathbf{G}})}{\|\mathbf{u}_i\|^2_{\boldsymbol{\Sigma}\mathbf{G}^2}}\mathbf{G}\mathbf{u}_i. \tag{63}$$

Eventually, by evaluating the weighted energies of both sides of the above (weighted by $\boldsymbol{\Sigma}$), we arrive at the fundamental weighted-energy conservation relation for

[6] For any column vector $\mathbf{x}$, the notation $\|\mathbf{x}\|^2_{\boldsymbol{\Sigma}}$ denotes the weighted squared Euclidean norm, i.e., $\|\mathbf{x}\|^2_{\boldsymbol{\Sigma}}=\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}$.

Please cite this article as: U.M. Al-Saggaf, et al., The $q$-Least Mean Squares algorithm, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.11.016

the $q$-LMS algorithm:

$$\|\tilde{\mathbf{w}}_{i+1}\|^2_{\boldsymbol{\Sigma}} + \frac{(e^{\boldsymbol{\Sigma}\mathbf{G}}_{ai})^2}{\|\mathbf{u}_i\|^2_{\boldsymbol{\Sigma}\mathbf{G}}} = \|\tilde{\mathbf{w}}_i\|^2_{\boldsymbol{\Sigma}} + \frac{(e^{\boldsymbol{\Sigma}\mathbf{G}}_{pi})^2}{\|\mathbf{u}_i\|^2_{\boldsymbol{\Sigma}\mathbf{G}}}. \tag{64}$$

Now, substituting the expression for the a posteriori error from (62) in (64) and taking the expectation on both sides of the above, with the aid of assumptions $\mathbf{A_1}$ and $\mathbf{A_2}$, to reach[7]

$$E[\|\tilde{\mathbf{w}}_{i+1}\|^2_{\boldsymbol{\Sigma}}] = E[\|\tilde{\mathbf{w}}_i\|^2_{\overline{\boldsymbol{\Sigma}}_2}] + \mu^2\sigma^2_\eta E[\|\mathbf{u}_i\|^2_{\boldsymbol{\Sigma}\mathbf{G}^2}], \tag{65}$$

where

$$\overline{\boldsymbol{\Sigma}}_2 = E[\boldsymbol{\Sigma}_2] = \boldsymbol{\Sigma} - 2\mu E[\mathbf{u}_i\mathbf{u}_i^T]\boldsymbol{\Sigma}\mathbf{G} + \mu^2 E[\|\mathbf{u}_i\|^2_{\boldsymbol{\Sigma}\mathbf{G}^2}\mathbf{u}_i\mathbf{u}_i^T]. \tag{66}$$

To proceed further, we use assumption **A3** which allows us to evaluate the input dependent moments appearing in (66). This gives $\overline{\boldsymbol{\Sigma}}_2$ a new look:

$$\overline{\boldsymbol{\Sigma}}_2 = \boldsymbol{\Sigma} - 2\mu\boldsymbol{\Sigma}\mathbf{G}\boldsymbol{\Lambda} + \mu^2\left[2\boldsymbol{\Sigma}\mathbf{G}^2\boldsymbol{\Lambda}^2 + \mathrm{Tr}(\boldsymbol{\Sigma}\mathbf{G}^2\boldsymbol{\Lambda})\boldsymbol{\Lambda}\right], \tag{67}$$

and the last moment appearing in (65) is found to be

$$E[\|\mathbf{u}_i\|^2_{\boldsymbol{\Sigma}\mathbf{G}^2}] = \mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{\Sigma}\mathbf{G}^2). \tag{68}$$

Now, defining the vectors $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_2$ comprising diagonal entries of matrices $\boldsymbol{\Sigma}$ and $\overline{\boldsymbol{\Sigma}}_2$, respectively, that is,

$$\boldsymbol{\sigma} \triangleq \mathrm{diag}(\boldsymbol{\Sigma}), \quad \text{and} \quad \boldsymbol{\sigma}_2 \triangleq \mathrm{diag}(\overline{\boldsymbol{\Sigma}}_2), \tag{69}$$

which allow us to relate $\boldsymbol{\sigma}$ with $\boldsymbol{\sigma}_2$ by the following relation:

$$\boldsymbol{\sigma}_2 = \mathbf{F}\boldsymbol{\sigma}, \tag{70}$$

where $\mathbf{F}$ is defined in (34).

Finally, using relations (69) and (70), the mean-square performance of the $q$-LMS algorithm can be shown to be governed by the recursion provided in (33).

## References

[1] A.H. Sayed, Fundamentals of Adaptive Filtering, Wiley-Interscience, New York, 2003.

[2] S. Haykin, Adaptive Filter Theory, 3rd edition, Prentice-Hall, Upper-Saddle River, NJ, 1996.

[3] B. Widrow, J.M. McCool, M.G. Larimore, C.R. Johnson, Stationary and nonstationary learning characteristics of the LMS adaptive filter, IEEE Proc. 64 (1976) 1151–1162.

[4] R.H. Kwong, E.W. Johnston, A variable stepsize LMS algorithm, IEEE Trans. Signal Process. 40 (1992) 1633–1642.

[5] R.W. Harris, D.M. Chabries, F.A. Bishop, A variable stepsize (VS) algorithm, IEEE Trans. Acoustic Speech Signal Process. 34 (1986) 499–510.

[6] T.J. Shan, T. Kailath, Adaptive algorithms with an automatic gain control feature, IEEE Trans. Acoust. Speech Signal Process. 35 (1988) 122–127.

[7] V.J. Methews, Z. Xie, A stochastic gradient adaptive filter with gradient adaptive step size, IEEE Trans. Signal Process. 41 (1993) 2075–2087.

[8] T. Aboulnasr, K. Mayyas, A robust variable step-size LMS type algorithm: analysis and simulation, IEEE Trans. Signal Process. vol. 45 (1997) 631–639.

[9] A.I. Sulyman, A. Zerguine, Convergence and steady-state analysis of a variable step-size nLMS algorithm, Signal Process. 83 (June) (2003) 1255–1273.

[10] J.I. Nagumo, A. Noda, A learning method for system identification, IEEE Trans. Autom. Control 12 (1967) 282–287.

[11] A. Zerguine, M.K. Chan, T.Y. Al-Naffouri, M. Moinuddin, C.F.N. Covan, Convergence and tracking analysis of a variable normalized LMF (XE-NLMF) algorithm, Signal Process.ing 89 (October (5)) (2008) 778–790.

[12] Y. Wei, S.B. Gefand, J.V. Krogmeier, Noise Constrained Least Mean Squares algorithm, IEEE Trans. Signal Process. 49 (September) (2001) 1961–1970.

[13] M. Moinuddin, A. Zerguine, A.U.H. Sheikh, Multiple access interference plus noise constrained least mean square (MNCLMS) algorithm for CDMA systems, IEEE Trans. Circuit Syst. 55 (October (9)) (2008) 2870–2883.

[14] A. Zerguine, M. Moinuddin, S.A.A. Imam, A noise constrained least mean fourth (NCLMF) adaptive algorithm, Signal Process. 91 (January (1)) (2011) 136–149.

[15] Y. Liu, W.K.S. Tang, Enhanced incremental LMS with norm constraints for distributed in-network estimation, Signal Process. 94 (January) (2014) 373–385.

[16] W. Liu, J.C. Principe, S. Haykin, Kernel Adaptive Filtering: A Comprehensive Introduction, Wiley & Sons, Hoboken, NJ, 2010.

[17] P. Bouboulis, S. Theodoridis, The complex Gaussian kernel LMS algorithm, in: International Conference on Artificial Neural Networks, September 2010, pp. 11–20.

[18] P. Bouboulis, S. Theodoridis, Extension of Wirtinger's calculus to reproducing Kernel Hilbert spaces and the complex Kernel LMS, IEEE Trans. Signal Process. 59 (3) (2011) 964–978.

[19] T. Ernst, The History of q-Calculus and a New-Method, U.U.D.M. Report 2000:16, Department of Mathematics, Uppsala University, Sweden, 2000.

[20] F.H. Jackson, On q-functions and a certain difference operator, Trans. R. Soc. Edinb. 46 (1908) 253–281.

[21] F.H. Jackson, On q-definite integrals, Q. J. Pure Appl. Math. 41 (1910) 193–203.

[22] J. Koekoev, A note on the q-derivative operator, J. Math. Anal. Appl. 176 (1993) 627–634.

[23] Aline C. Soterroni, Roberto L. Galski, Fernando M. Ramos, The q-gradient method for global optimization, J. Math. (2013). Q2



[7] Here, we have used the property that $E[\|\mathbf{x}\|^2_{\boldsymbol{\Sigma}}] = E[\|\mathbf{x}\|^2_{E[\boldsymbol{\Sigma}]}]$ [1].