# A NEW QR-BASED BLOCK LEAST MEAN SQUARES (QR-BLMS) ALGORITHM FOR ADAPTIVE PARAMETER ESTIMATION

*Yang Xin-xing* and *S. C. Chan*
Department of Electrical and Electronic Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong

## ABSTRACT

This paper proposes a new family of QR-based block LMS (QR-BLMS) algorithms for adaptive parameter estimation. It extends the QR-based LMS (QR-LMS) algorithm by handling a block of data vectors at a time instead of one single input vector. Moreover, it is shown that when there is only one new input vector in the block and the others are obtained from previous time instants, this QR-BLMS algorithm yields a new QR-based implementation of the well-known affine projection algorithm (APA). Simulation results for an acoustic echo canceller showed that the QR-BLMS and QR-APA algorithms perform better than the QR-LMS algorithm.

## 1. INTRODUCTION

Efficient recursive least squares (RLS) algorithms using the QR decomposition (QRD) is well known for their good numerical property [1]-[3], because the condition number of the system is much lower than that of the input correlation matrix. The QRD-based algorithms usually consist of the following two separate parts: 1) recursive updating of the triangular matrix and 2) backsolving of the filter parameters. Although the matrix-updating step can be efficiently performed with $O(N)$ arithmetic operations for single input adaptive filtering applications, the backsolving step still requires $O(N^2)$ operation. Here, $N$ is the length of the adaptive transversal filter. Therefore, in applications where the filter parameters have to be computed, the entire algorithm still requires $O(N^2)$ arithmetic operations. Recently, an approximate QR-based LS (A-QR-LS) fast adaptive parameter estimation algorithm with $O(N)$ complexity was proposed by Liu [4], where a special structured matrix is used to approximate the triangular matrix in the QRD. Because of this structure, the triangularzation and backsolving steps can be combined together yielding an algorithm of $O(N)$ arithmetic complexity. Based on this algorithm, a similar QR-based least mean squares (QR-LMS) algorithm was proposed in [5]. The QR-LMS algorithm is mathematically equivalent to the optimum nonlinearly modified least mean squares (ONM-LMS) algorithm in an infinite precision environment. However, the use of Householder transform in solving for the filter parameters greatly improves numerical properties and convergence speed of the QR-LMS algorithm over the ONM-LMS algorithm. The tracking performance of the QR-LMS algorithm for time-varying parameters is also significantly faster. In this paper, we propose a new family of QR-based block LMS (QR-BLMS) adaptive parameter estimation algorithms that handle block instead of one vector at a time. This new algorithm has a faster convergence speed than the QR-LMS algorithm. Moreover, we show that when there is only one new input vector in the block and the others are obtained from previous time instants, this QR-BLMS algorithm corresponds to a new QR-based implementation of the well-known affine projection algorithm. Thanks to the Householder transformation, this QR-APA algorithm has better numerical property and convergence speed. The paper is organized as follows: the QR-LMS is briefly reviewed in Section 2. The proposed QR-BLMS algorithm is described in Section 3. Simulation results and comparison to other algorithms are given in Section 4. Finally, conclusions are drawn in Section 5.

## 2. QR-BASED LMS ALGORITHM

In this section, the QR-LMS [5] algorithm will be briefly reviewed. Consider the estimation of the $N$-dimensional parameter vector $\boldsymbol{\theta}$ for the following linear model

$$d_n = \boldsymbol{x}_n^T\boldsymbol{\theta} + v(n)\ , \tag{1}$$

where $(\cdot)^T$ denotes matrix transpose, $d_n$ and $\boldsymbol{x}_n^T = [x_n(1), x_n(2), \ldots, x_n(N)]$ are the desired (observed) signal and input vectors, respectively, and $v(n)$ is an additive white Gaussian noise sequence with zero mean. In the ONM-LMS algorithm, the weight update equation for the well-known LMS algorithm is

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + \mu[d_n - \boldsymbol{x}_n^T(N)\hat{\boldsymbol{\theta}}_{n-1}]\boldsymbol{x}_n(N)\ , \tag{2}$$

where $\hat{\boldsymbol{\theta}}_n$ be the estimated parameter vector at time instant $n$,

$$\mu = 1/(w_n^2 + \boldsymbol{x}_n^T\boldsymbol{x}_n) \tag{3}$$

is the step size parameter, and $w_n$ is a weighting factor. By substituting $\mu$ in (3) into (2), we get

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + \frac{d_n - \boldsymbol{x}_n^T(N)\hat{\boldsymbol{\theta}}_{n-1}}{w_n^2 + \boldsymbol{x}_n^T\boldsymbol{x}_n}\boldsymbol{x}_n(N). \tag{4}$$

Equation (4) can be reorganized as follows

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} - \frac{\boldsymbol{x}_n\boldsymbol{x}_n^T\hat{\boldsymbol{\theta}}_{n-1}}{w_n^2 + \boldsymbol{x}_n^T\boldsymbol{x}_n} + w_n^{-2}d_n\hat{\boldsymbol{\theta}}_n - \frac{w_n^{-2}d_n\boldsymbol{x}_n\boldsymbol{x}_n^T\boldsymbol{x}_n}{w_n^2 + \boldsymbol{x}_n^T\boldsymbol{x}}$$

$$= w_n^{-2}\left(\boldsymbol{I} - \frac{\boldsymbol{x}_n\boldsymbol{x}_n^T}{\boldsymbol{w}_n^{-2} + \boldsymbol{x}_n^T\boldsymbol{x}_n}\right)(w_n^2\hat{\boldsymbol{\theta}}_{n-1} + d_n\boldsymbol{x}_n)$$

$$= \left\{\begin{bmatrix} w_n\boldsymbol{I} & \boldsymbol{x}_n \end{bmatrix}\begin{bmatrix} w_n\boldsymbol{I} \\ \boldsymbol{x}_n^T \end{bmatrix}\right\}^{-1}\begin{bmatrix} w_n\boldsymbol{I} & \boldsymbol{x}_n \end{bmatrix}\begin{bmatrix} w_n\hat{\boldsymbol{\theta}}_{n-1} \\ d_n \end{bmatrix}$$

$$= (\boldsymbol{\varphi}_n^T\boldsymbol{\varphi}_n)^{-1}\boldsymbol{\varphi}_n^T\boldsymbol{b}_n, \tag{5}$$

where $\boldsymbol{I}$ is an $N \times N$ identity matrix, $\boldsymbol{\varphi}_n = \begin{bmatrix} w_n\boldsymbol{I} & \boldsymbol{x}_n \end{bmatrix}^T$, and $\boldsymbol{b}_n = \begin{bmatrix} w_n\hat{\boldsymbol{\theta}}_{n-1}^T & d_n \end{bmatrix}^T$. Note, in deriving (5), the matrix inversion lemma has been used. Therefore, the OMN-LMS estimate of $\hat{\boldsymbol{\theta}}_n$ is also the solution of the normal equation in (5). This in turns is the least square (LS) solution of the following overdetermined equation

$$\boldsymbol{\varphi}_n\boldsymbol{\theta}_n \approx \boldsymbol{b}_n. \tag{6}$$

Let $\hat{\theta}_{n-1}(i)$ and $x_n(i)$ be respectively the $i$-th elements of $\hat{\boldsymbol{\theta}}_{n-1}$ and $\boldsymbol{x}_n$, and $\boldsymbol{\beta}_n = \begin{bmatrix} \boldsymbol{\theta}_n^T & 1 \end{bmatrix}^T$. Then, we can rewrite (6) as

$$\begin{bmatrix} w_n & & & & -w_n\hat{\theta}_{n-1}(1) \\ & w_n & & \boldsymbol{0} & -w_n\hat{\theta}_{n-1}(2) \\ \boldsymbol{0} & & \ddots & & \vdots \\ & & & w_n & -w_n\hat{\theta}_{n-1}(N) \\ x_n(1) & x_n(2) & \cdots & x_n(N) & -d_n \end{bmatrix} \cdot \boldsymbol{\beta}_n \approx \boldsymbol{0}. \tag{7}$$

Hence, $\hat{\boldsymbol{\theta}}_n$ can also be obtained by solving eqn. (7). Because of the special structure of the matrix on the left hand side of (7), it can be solved efficiently using the numerically more stable QRD such as Householder reflection or Givens rotation. The QR-LMS in [5] employs the Householder transformation and combines the updating of this matrix and the back-solving process together. The arithmetic complexity of the resulting algorithm is only $O(N)$. More recently, the authors have proposed an improvement QR-LMS algorithm using square-free Givens rotations [7]. More precisely, it was observed that the Givens rotation is more efficient than the Householder reflection if the input vector is processed one at a time. The arithmetic complexity can be reduced by a factor of two. Furthermore, the Givens rotation-based QR-LMS algorithm allows us to derive a square-root free version similar to the square-root free RLS algorithm of [9]. Simulation results in [5] and [7] showed that the QR-LMS has a better numerical accuracy and tracking speed than the OMN-LMS algorithm. In the next section, we shall develop a new QR-based block LMS (QR-BLMS) algorithm with better convergence speed.

## 3. QR-BASED BLOCK LMS

In the QR-LMS algorithm, a new data vector is appended to the data matrix one at a time (eqn. 7). Here, we extend this algorithm to handle multiple new data vectors and/or more data vectors from previous time instants at each iteration. Under this situation, the QRD performed on the matrix in (7) resembles more closely the operations in the conventional QRD-based RLS algorithm for the given block of data input. However, the triangular factor will be retained in the RLS algorithm, while the QR-BLMS algorithm only retains a structure similar to (7) so as to combine the triangularzation and backsolving processes together to yield an algorithm with O($N$) complexity. As the block dimension increases, the QRD operations in solving (7) can make better use of the data information to improve the convergence speed. This is very similar to the situation between the Affine Projection algorithm (APA) [8] and the LMS algorithm. In fact, we shall show that our QR-BLMS algorithm reduces to the normalized APA if the block only consists of one new input data vector, while the others are obtained from previous time instants. The resulting algorithm, which inherences the excellent numerical property of the Householder or Givens rotation, is called the QR-based APA (QR-APA) algorithm. Without loss of generality, assume that $L$ (>0) rows of new data are appended to the data matrix at a time. Then, equation (7) becomes

$$\boldsymbol{D}_N \cdot \boldsymbol{\beta}_n = \begin{bmatrix} w_n & & & & -w_n\hat{\theta}_{n-1}(1) \\ & w_n & & \boldsymbol{0} & -w_n\hat{\theta}_{n-1}(2) \\ \boldsymbol{0} & & \ddots & & \vdots \\ & & & w_n & -w_n\hat{\theta}_{n-1}(N) \\ \boldsymbol{x}_n(1) & \boldsymbol{x}_n(2) & \cdots & \boldsymbol{x}_n(N) & -\boldsymbol{d}(n) \end{bmatrix} \cdot \boldsymbol{\beta}_n \approx \boldsymbol{0} \tag{8}$$

where $\boldsymbol{x}_n(k) = [x_n(k), x_{n-1}(k), \cdots, x_{n-L+1}(k)]^T$, for $k = 1,2,\cdots,N$, and $\boldsymbol{d}(n) = [d_n, d_{n-1}, \cdots, d_{n-L+1}]^T$. (8) can also be rewritten as

$$\boldsymbol{\Phi}_n\boldsymbol{\theta}_n = \boldsymbol{c}_n \tag{9}$$

where $\boldsymbol{\Phi}_n = [w_n\boldsymbol{I} \quad \boldsymbol{X}_n]^T$, $\boldsymbol{c}_n = [w_n\hat{\boldsymbol{\theta}}_{n-1} \quad (\boldsymbol{d}(n))^T]^T$, (10) and $\boldsymbol{X}_n = [\boldsymbol{x}_n(1), \boldsymbol{x}_n(2), \cdots, \boldsymbol{x}_n(N)]^T$.

To show that (8) and (9) actually corresponds to the normalized block LMS algorithm, we first note that the LS solution of (9) is $(\boldsymbol{\Phi}_n^T \boldsymbol{\Phi}_n)^{-1} \boldsymbol{\Phi}_n \boldsymbol{c}_n$. Further, using (10), one gets

$$
\begin{aligned}
\hat{\boldsymbol{\theta}}_n &= (\boldsymbol{\Phi}_n^T \boldsymbol{\Phi}_n)^{-1} \boldsymbol{\Phi}_n \boldsymbol{c}_n \\
&= \left\{ [w_n \boldsymbol{I} \quad \boldsymbol{X}_n] \begin{bmatrix} w_n \boldsymbol{I} \\ \boldsymbol{X}_n^T \end{bmatrix} \right\}^{-1} [w_n \boldsymbol{I} \quad \boldsymbol{X}_n] \begin{bmatrix} w_n \hat{\boldsymbol{\theta}}_{n-1} \\ \boldsymbol{d}(n) \end{bmatrix} \\
&= w_n^{-2} (\boldsymbol{I} - \boldsymbol{X}_n (\boldsymbol{X}_n^T \boldsymbol{X}_n + w_n^2 \boldsymbol{I})^{-1} \boldsymbol{X}_n^T)(w_n^2 \hat{\boldsymbol{\theta}}_{n-1} + \boldsymbol{X}_n \boldsymbol{d}(n)) \cdot \qquad (11) \\
&= \hat{\theta}_{n-1} - \boldsymbol{X}_n (\boldsymbol{X}_n^T \boldsymbol{X}_n + w_n^2 \boldsymbol{I})^{-1} \boldsymbol{X}_n^T \hat{\boldsymbol{\theta}}_{n-1} + w_n^{-2} \boldsymbol{X}_n \boldsymbol{d}(n) \\
&\quad - w_n^{-2} \boldsymbol{X}_n (\boldsymbol{X}_n^T \boldsymbol{X}_n + w_n^2 \boldsymbol{I})^{-1} \boldsymbol{X}_n^T \boldsymbol{X}_n \boldsymbol{d}(n) \\
&= \hat{\theta}_{n-1} + \boldsymbol{X}_n (\boldsymbol{X}_n^T \boldsymbol{X}_n + w_n^2 \boldsymbol{I})^{-1} (\boldsymbol{d}(n) - \boldsymbol{X}_n^T \hat{\boldsymbol{\theta}}_{n-1}).
\end{aligned}
$$

(11) is recognized as a normalized block LMS algorithm. (11) can be shown to converge if 0<w<1. However, due to page limitation, the mean convergence of (11) is omitted. Like the QR-LMS algorithm, solving the equivalent system in (8) using the Householder-based QRD leads to better numerical properties. We call this algorithm the QR-BLMS algorithm. When only one new data vector is being added to the data block $\boldsymbol{X}_n$ while the other vectors in the block are obtained from previous time instants, (11) reduces to the well-known affine projection algorithm (APA). If its equivalent form in (8) is solved using Householder transformation, then we obtain a new QR-based APA algorithm, called QR-APA algorithm. Like QR-BLMS, the QR-APA algorithm is expected to offer better numerical properties and fast tracking speed. Since the derivation of the QR-APA is similar to that of QR-BLMS, only details for the latter will be given below. Consider the matrix $\boldsymbol{D}_N$ in (8). It can be transformed into an upper triangular matrix using a series of Householder matrices $\boldsymbol{H}_n = \boldsymbol{H}_n(N)\boldsymbol{H}_n(N-1)\cdots\boldsymbol{H}_n(1)$

$$\boldsymbol{D}_N^{(i+1)}(n) = \boldsymbol{H}_n(i)\boldsymbol{D}_N^{(i)}(n), \quad i = 1,2,\cdots,N \qquad (12)$$

where $\boldsymbol{D}_N^{(1)} = \boldsymbol{D}_N$, and $\boldsymbol{D}_N^{(N+1)}$ is the desired upper triangular matrix; $\boldsymbol{H}_n(i)$ is a symmetric and orthogonal matrix given by $\boldsymbol{H}_n(i) = \boldsymbol{I} - \boldsymbol{u}^{(i)}(\boldsymbol{u}^{(i)})^T / \sigma_i$. $\boldsymbol{H}_n(i)$ is chosen such that $\boldsymbol{x}_n^{(i+1)}(i) = \boldsymbol{0}$, where $\boldsymbol{x}_n^{(i+1)}(i)$ is the element of matrix $\boldsymbol{D}_N^{(i+1)}(n)$ in row *N* and column *i*. The corresponding parameters of $\boldsymbol{H}_n(i)$ are given: $\alpha_i = \sqrt{w_n^2 + \|\boldsymbol{x}_n^{(i)}(i)\|^2}$, $\sigma_i = \alpha_i(\alpha_i + |w_n|)$

$$
u_j^{(i)} = \begin{cases} w_n + \mathrm{sign}(w_n)\alpha_i = \eta_i & \text{if } j = i \\ \boldsymbol{x}_n^{(i)}(i) & \text{if } j = N+1, \\ 0 & \text{otherwise} \end{cases}
$$

$$i = 1,2,\cdots,N. \qquad (13)$$

Therefore, (12) can be rewritten as

$$\boldsymbol{D}_N^{(i+1)}(n) = \boldsymbol{D}_N^{(i)}(n) - \boldsymbol{u}^{(i)} \boldsymbol{q}_i^T / \sigma_i$$

where $\boldsymbol{q}_i^T = (\boldsymbol{u}^{(i)})^T \boldsymbol{D}_N^{(i)}(n) = [0,\cdots,0,\tau_i,\tau_{i+1},\cdots,\tau_{N+1}]$ (14)

$$
\tau_j = \begin{cases} \eta_i w_n + \|\boldsymbol{x}_n^{(i)}(i)\|^2 & \text{if } j = i \\ (\boldsymbol{x}_n^{(i)}(i))^T \boldsymbol{x}_n^{(i)}(j) & \text{if } j = i+1,\cdots,N \\ \eta_i(-w_n \hat{\boldsymbol{\theta}}_{n-1}) + (\boldsymbol{x}_n^{(i)}(i))^T \boldsymbol{d}^{(i)}(n) & \text{if } j = N+1 \end{cases} \qquad (15)
$$

(13) to (15) yields the following recursive Househoulder triangularization algorithm:

*For* $i = 1,2,\cdots,N$ *Loop*

$\alpha_i = \sqrt{w_n^2 + \|\boldsymbol{x}_n^{(i)}(i)\|^2}$, $\sigma_i = \alpha_i(\alpha_i + |w_n|)$

$\eta_i = w_n + \mathrm{sign}(w_n)\alpha_i$, $\eta_i' = \eta_i / \sigma_i$

$\mu_i = \boldsymbol{x}_n^{(i)}(i) / \sigma_i$

$r_{i,i}(n) = -\mathrm{sign}(w_n)\alpha_i$

$d_{i,N+1} = -w_n \hat{\boldsymbol{\theta}}_{n-1} - \eta_i'(\eta_i(-w_n \hat{\boldsymbol{\theta}}_{n-1}) + (\boldsymbol{x}_n^{(i)}(i))^T \boldsymbol{d}^{(i)}(n))$

$\boldsymbol{d}^{(i+1)}(n) = \boldsymbol{d}^{(i)}(n) - \mu_i(\eta_i(-w_n \hat{\boldsymbol{\theta}}_{n-1}) + (\boldsymbol{x}_n^{(i)}(i))^T \boldsymbol{d}^{(i)}(n))$

*For* $j = i+1, i+2, \cdots, N$ *Loop*

$d_{i,j} = -\eta_i'(\boldsymbol{x}_n^{(i)}(i))^T \boldsymbol{x}_n^{(i)}(i)$

$\boldsymbol{x}_n^{(i+1)}(j) = (\boldsymbol{I} - \dfrac{\boldsymbol{x}_n^{(i)}(i)(\boldsymbol{x}_n^{(i)}(i))^T}{\sigma_i})\boldsymbol{x}_n^{(i)}(j)$

*End of Loop j*

*End of Loop i* (16)

$d_{i,j}$ and $r_{i,i}(n)$ denote the element of the upper triangular matrix transformed from matrix $\boldsymbol{D}_N(n)$. Let $\rho_i = \boldsymbol{I} - \boldsymbol{x}_n^{(i)}(i)(\boldsymbol{x}_n^{(i)}(i))^T / \sigma$ and $\pi_i = \prod_{j=i}^{1} \rho_j$. Then, by rewriting $\boldsymbol{x}_n^{(i+1)}(j)$ in (16) as

$$\boldsymbol{x}_n^{(i+1)}(j) = \pi_i \boldsymbol{x}_n(j), \; j = i+1, i+2, \cdots, N. \qquad (17)$$

So we can rewrite $\rho_i = \boldsymbol{I} - \pi_{i-1}\boldsymbol{x}_n(i)(\pi_{i-1}\boldsymbol{x}_n(i))^T / \sigma_i$. Further, by defining $\delta_i = -\mathrm{sign}(w_n)(\cdot\pi_{i-1}\boldsymbol{x}_n(i))^T \pi_{i-1} / \sigma_i$, we get the desired QR-BLMS algorithm in Table 1. The arithmetic complexity/iteration of this algorithm is $O(L^2 N)$. Because the QR-BLMS is a block algorithm and it produces an output every *L* cycles, its complexity is $O(LN)$, as compared with $O(L^2 N)$ for the QR-APA, which performs similar operations every cycle. Similar algorithm using Givens rotation can be derived.

## 4. EXPERIMENTAL RESULTS

The performances of the various algorithms are evaluated using an acoustic echo cancellation problem. The input signal is an artificially generated speech signal, which is modeled as a 5-th order AR model characterized by poles at: $z_1 = 0.5$, $z_2$, $z_3 = 0.85 e^{\pm j\pi/3}$, $z_4$ and $z_5 = 0.7 e^{\pm j 2\pi/3}$. This AR model is driven by a white noise with zero mean and unit variance. The acoustic path of the echo is modeled as a linear time invariant system using an exponentially weighted model of order 60. The adaptive filters are assumed to have an order of 100, which is larger than the actual length of the acoustic path. The error norms for QR-BLMS, QR-APA and QR-LMS are shown in Fig. 1. Here the block size $L$ is equal to two and three. Fig. 1 shows the error norm of the parameters as a function of time and it is obtained by averaging over 100 Monte Carlo simulations. It can be seen that as the block dimension increases, so does the convergence speed of the parameters and the QR-BLMS and QR-APA perform much better than QR-LMS. Note, although the QR-BLMS is a block algorithm, its performance is very close to the QR-APA algorithm of $L$ times more complexity. This substantiates the usefulness of the proposed methods.

## 5. CONCLUSIONS

A new family of QR-BLMS algorithms for adaptive parameter estimation is presented. It extends the QR-LMS algorithm by handling a block of data vectors at a time instead of one single input vector. It is shown that when there is only one new input vector in the block and the others are obtained from previous time instants, this QR-BLMS algorithm yields a new QR-based implementation of the well-known affine projection algorithm (APA). Simulation results for an acoustic echo canceller showed that the QR-BLMS and QR-APA perform much better than QR-LMS algorithm.

## 6. REFERENCES


[1] J. G. McWhirter, "Recursive least squares minimization using a systolic array," in *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 431, 1983, pp. 105-112.

[2] J. M. Cioffi, "A fast adaptive ROTOR's RLS algorithm," *IEEE Trans. ASSP*, vol. 38, 1990, pp. 631-653.

[3] D. T. M. Slock and T. Kailath, "Numerically stable fast recursive least squares transversal filters," *Proc. IEEE ICASSP'*1988, pp. 1369-1372.

[4] Z. S. Liu, "QR methods of *O*(N) complexity in adaptive parameter estimation," *IEEE Trans. SP*, vol. 43, 1995, pp. 720-729.

[5] Z. S. Liu and J. Li, "A QR-based least mean squares algorithm for adaptive parameter estimation," *IEEE Trans. CAS-II*, vol 45, 1998, pp. 321-329.

[6] M. Rupp, "Rursting in the LMA algorithm," *IEEE Trans. Signal Processing*, vol. 43, pp. 2414–2417, Oct. 1995.

[7] S. C. Chan and X. X. Yang, "Improved approximate QR-LS algorithms for adaptive filtering," submitted to *IEEE Trans. CAS-II*, AUG. 2002.

[8] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan*, vol. 67-A, No. 5, 1984.

[9] W. M. Gentleman and H. T. Kung, " Matrix triangularization by systolic array," in *Proc. SPIE Int. Soc. Opt. Eng.*, 1981, vol. 298, pp. 298-303.


*Upper Triangular Algorithm*

$\pi_0 = 1$

*For* $i = 1, 2, \cdots, N$ *Loop*

$$\alpha_i = \sqrt{w_n^2 + \|\pi_{i-1}\boldsymbol{x}_n(i)\|^2}, \quad \sigma_i = \alpha_i(\alpha_i + |w_n|)$$

$$\rho_i = \boldsymbol{I} - \pi_{i-1}\boldsymbol{x}_n(i)(\pi_{i-1}\boldsymbol{x}_n(i))/\sigma_i, \quad \delta_i = -\text{sign}(w_n)(\cdot\pi_{i-1}\boldsymbol{x}_n(i))^T \pi_{i-1}/\sigma_i$$

$$\pi_i = \rho_i\pi_{i-1}, \; \eta_i = w_n + \text{sign}(w_n)\alpha_i, \; \eta_i' = \eta_i/\sigma_i, \; \mu_i = \pi_{i-1}\boldsymbol{x}_n(i)/\sigma_i$$

$$r_{i,i}(n) = -\text{sign}(w_n)\alpha_i$$

$$d_{i,N+1} = -w_n\hat{\boldsymbol{\theta}}_{n-1} - \eta_i'(\eta_i(-w_n\hat{\boldsymbol{\theta}}_{n-1}) + (\boldsymbol{x}_n^{(i)}(i))^T \boldsymbol{d}^{(i)}(n)$$

$$\boldsymbol{d}^{(i+1)}(n) = \boldsymbol{d}^{(i)}(n) - \mu_i(\eta_i(-w_n\hat{\boldsymbol{\theta}}_{n-1}) + (\boldsymbol{x}_n^{(i)}(i))^T \boldsymbol{d}^{(i)}(n))$$

*End of Loop i*

*Backsolving Algorithm*

$\gamma_N = 0, s_N(n) = -d_{N,N+1}$

$\theta_n(N) = s_N(n) / r_{i,i}(n)$

*For* $i = N-1, N-2, \cdots, 1$ *Loop*

$$\gamma_i = \gamma_{i+1} + \boldsymbol{x}_n(i+1)\theta_n(i+1), \quad s_i(n) = -d_{i,N+1} - \delta_i\gamma_i$$

$$\theta_n(i) = s_i(n) / r_{i,i}(n)$$

*End of Loop i*

Table 1. QR-BLMS Algorithm.

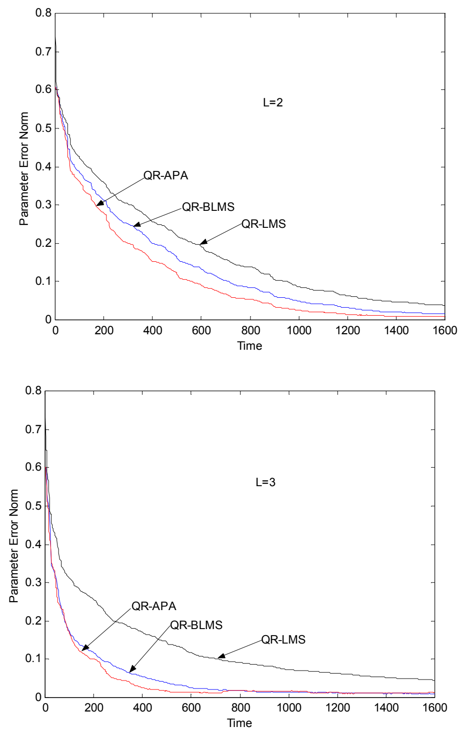


Fig. 1 Parameter Error Norm for QR-BLMS, QR-APA and QR-LMS ($L$=2,3). $w = 0.001$.