

A NEW FAMILY OF APPROXIMATE QR-LS ALGORITHMS FOR ADAPTIVE FILTERING

Y. Zhou and S. C. Chan

yizhou@eee.hku.hk, scchan@eee.hku.hk

Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong

ABSTRACT

This paper proposes a new family of approximate QR-based least squares (LS) adaptive filtering algorithms called p -TA-QR-LS algorithms. It extends the TA-QR-LS algorithm [6] by retaining different number of diagonal plus off-diagonals (denoted by an integer p) of the triangular factor of the augmented data matrix. For $p=1$ and N , it reduces respectively to the TA-QR-LS and the QR-RLS algorithms. It not only provides a link between the QR-LMS-type and the QR-RLS algorithms through a well-structured family of algorithms, but also offers flexible complexity-performance tradeoffs in practical implementation. These results are verified by computer simulation and the mean convergence of the algorithms is also analyzed.

1. INTRODUCTION

Adaptive filtering is widely used in communications, control, and many other applications. Many adaptive filtering algorithms have been proposed [1]. The recursive least square (RLS) algorithm has a fast convergence speed but a very high arithmetic complexity of $O(N^2)$ (where N is the number of taps of the adaptive filter). A numerically more preferred implementation is based on the QR decomposition, which usually exhibits better numerical property due to the reduced condition number [2]. Moreover, by exploiting the shifting property in adaptive filtering, the arithmetic complexity for updating the triangular factor can be further reduced to $O(N)$. However, the back-solving step still requires $O(N^2)$ operations. In order to reduce the complexity of the back-solving step in the QRD, Liu proposed an approximate QR-LS algorithm (A-QR-LS) [3] and a related QR-LMS algorithm [4] based on Householder transformation [2], which

combines the recursive updating of the triangular matrix and the back-solving of the parameters. Another related QRD block algorithm proposed by Bhourri et al [5] approximates the upper triangular factor by a series of block upper triangular matrices and includes [3] as its special case when their sizes are reduced to one. Like the fast Newton algorithm [8], the goal is to approximate the covariance matrix or its factor to reduce the arithmetic complexity. Recently, Chan and Yang [6] showed that the A-QR-LS algorithm in [3] is equivalent to an element-wise normalized LMS algorithm with time-varying step-sizes. It reduces to the QR-LMS algorithm in [4] when all the normalization constants are chosen as the Euclidean norm of the input signal vector. Based on these observations, they proposed a transform domain A-QR-LS (TA-QR-LS) algorithm based on square-root free (SRF) Givens rotation method [2] which possesses both fast convergence speed and a reduced arithmetic complexity. It was also shown in [7] that a block version of the QR-LMS algorithm represents a fast implementation of the fast affine projection algorithm (APA). In this paper, we propose a new family of QR-based algorithms called the p -TA-QR-LS algorithm. By retaining different number of diagonal plus off-diagonals (denoted by $p, 1 \leq p \leq N$) during the QRD, the triangular factor of the data matrix can be approximated to different levels according to the value of p . Coupling with the back-solving step as in [6], we obtain a new family of QR-based algorithms with different complexity-performance tradeoff, parameterized by the values of the integer p . For $p=1$, it reduces to the TA-QR-LS algorithm in [6]. When $p=N$, the conventional QR-RLS algorithm is obtained. Other values of p generate a series of new QR-based algorithms with complexity-performance tradeoff between these two well-known families of algorithms. It thus provides a direct link between the QR-LMS-type and the QR-RLS algorithms through a uniformly-structured family of algorithms parameterized by p . The proposed algorithm differs from the block algorithm in [5] in how the triangular factor is approximated and hence the solution method. It can be viewed as the generalizations of the QR-LMS and A-QR-LS algorithms in filling their

gaps with the QR-RLS algorithm. Moreover, we shall that the resulting algorithms with $p < N$ is a LMS-type algorithm and they converge in the mean. Likewise, their performances can further be improved using transformation as in the TA-QR-LS algorithm. The rest of this paper is organized as follows: the TA-QR-LS algorithm in [6] is briefly reviewed in section 2. The proposed p -TA-QR-LS algorithm and its mean convergence analysis are presented in section 3. In section 4, experimental results and comparisons are given. Finally, conclusions are drawn in section 5.

2. THE TA-QR-LS ALGORITHM

Consider the linear estimation of a parameter vector θ° with observation $d(j)$ and input $\mathbf{x}_N(j) = [x_1(j), x_2(j), \dots, x_N(j)]^T$ given by the model $d(j) = \mathbf{x}_N^T(j)\theta^\circ + \eta(j)$, where $\eta(j)$ is a zero mean, additive white Gaussian noise sequence. In least squares estimation, the time averaged squares error is minimized:

$$\xi_N(n) = \sum_{j=0}^n \lambda^{n-j} |e(j)|^2 \quad (1)$$

where $e(j) = d(j) - \mathbf{x}_N^T(j)\theta(n)$, $\theta(n)$ is the estimated parameter vector at time n , and λ is a constant forgetting factor with a value between 0 and 1. (1) can also be written more compactly in matrix form as

$$\xi_N(n) = \mathbf{e}^T(n)\mathbf{W}^2(n)\mathbf{e}(n) = \|\mathbf{W}(n)\mathbf{e}(n)\|^2 \quad (2)$$

where $\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}_N(n)\theta(n)$, $\mathbf{d}(n) = [d(0), d(1), \dots, d(n)]^T$, $\mathbf{W}(n) = \text{diag}(\sqrt{\lambda^n}, \sqrt{\lambda^{n-1}}, \dots, \sqrt{\lambda}, 1)$ is a diagonal weighting matrix, and $\mathbf{X}_N(n) = [\mathbf{x}_N(0), \mathbf{x}_N(1), \dots, \mathbf{x}_N(n)]^T$. The QR-LS (or QR-RLS in this paper) method [6] is frequently used to solve for the optimum value of $\theta(n)$ because of its superior numerical property. The QR-RLS method [6] is summarized below in Table 1.

<p>1. Given the augmented data matrix</p> $\mathbf{D}_N(n-1) = \mathbf{W}(n-1)[\mathbf{X}_N(n-1) \quad \mathbf{d}(n-1)]$ <p>and its QRD at time $(n-1)$:</p> $\mathbf{D}_N^*(n-1) = \mathbf{Q}(n-1)\mathbf{D}_N(n-1) = \begin{bmatrix} \mathfrak{R}_N(n-1) & \hat{\mathbf{d}}_N(n-1) \\ 0 & \mathbf{c}_{n-1} \end{bmatrix}$ <p>where $\mathbf{Q}(n-1)$ and $\mathfrak{R}_N(n-1)$ are unitary and upper triangular matrices, respectively.</p> <p>2. (QRD) Form the new augmented data matrix</p> $\mathbf{D}_N(n) = \mathbf{W}(n)[\mathbf{X}_N(n) \quad \mathbf{d}(n)] = \begin{bmatrix} \sqrt{\lambda}\mathbf{D}_N^*(n-1) \\ \psi_{N+1}^T(n) \end{bmatrix}$ <p>where $\psi_{N+1}(n) = [\mathbf{x}_N^T(n) \quad d(n)]^T$. Get the new QRD by Givens rotations or Householder reflections as</p> $\mathbf{Q}^{(N)}(n) \cdots \mathbf{Q}^{(1)}(n)\mathbf{Q}'(n)\mathbf{D}_N(n) = \begin{bmatrix} \mathfrak{R}_N(n) & \hat{\mathbf{d}}_N(n) \\ 0 & \mathbf{c}_n \\ 0^T & d^{N+1}(n) \end{bmatrix}$ <p>3. (Back-solving) Solve the triangular system $\mathfrak{R}_N(n)\theta(n) = \hat{\mathbf{d}}_N(n)$ for</p>
--

the LS estimate $\theta(n)$ at time n by back-substitution:

$$\theta_N(n) = [r_{N,N+1}(n)]/r_{N,N}(n)$$

$$\theta_i(n) = [r_{i,N+1}(n) - \sum_{j=i+1}^N r_{i,j}(n)\theta_j(n)]/r_{i,i}(n), \quad i = N-1, \dots, 1$$

where $r_{i,j}$ and $r_{i,N+1}$ are the corresponding elements in $\mathfrak{R}_N(n)$ and $\hat{\mathbf{d}}_N(n)$. $\theta_i(n)$ is the i -th element of $\theta(n)$.

Table 1. QR-RLS algorithm.

In [4], Liu proposed an approximate QR-LS (A-QR-LS) algorithm with an arithmetic complexity of $O(N)$ by approximating the upper triangular as a diagonal matrix, which simplifies the QRD and the back substitution above. More precisely, the quantities inside the square bracket in step 3 of table 1 are computed from $\theta_i(n-1)$ and are denoted by:

$$\begin{aligned} s_N(n-1) &= r_{N,N+1}(n-1), \\ s_i(n-1) &= [r_{i,N+1}(n-1) - \sum_{j=i+1}^N r_{i,j}(n-1)\theta_j(n-1)] \\ & \quad i = N-1, N-2, \dots, 1, \end{aligned} \quad (3)$$

or equivalently:

$$r_{i,i}(n-1)\theta_i(n-1) = s_i(n-1), \quad i = 1, \dots, N. \quad (4)$$

Given the values of $s_i(n-1)$ and $r_{i,i}(n-1)$, (4) together with the relationship $d(n) = \mathbf{x}_N^T(n)\theta^\circ + \eta(n)$ can be viewed as a system of linear equations in the variable $\theta(n)$:

$$\begin{aligned} w \cdot r_{i,i}(n-1)\theta_i(n) &= w \cdot s_i(n-1), \quad i = 1, \dots, N, \\ \mathbf{x}_N^T(n)\theta(n) &= d(n), \end{aligned} \quad (5)$$

where w is the square root of the forgetting factor. (5) can also be written in matrix form as:

$$\Phi(n)\theta(n) = \mathbf{b}(n), \quad (6)$$

where $\Phi(n) = \begin{bmatrix} w\mathbf{D}(n-1) \\ \mathbf{x}_N^T(n) \end{bmatrix}$, $\mathbf{b}(n) = \begin{bmatrix} w\mathbf{D}(n-1)\theta(n-1) \\ d(n) \end{bmatrix}$,

$$\mathbf{D}(n-1) = \text{diag}\{r_{1,1}(n-1), \dots, r_{N,N}(n-1)\}.$$

Therefore, (6) can be solved by computing the QRD of $\Phi(n)$, which actually works with the following appended matrix:

$$\tilde{\mathbf{D}}(n) = \begin{bmatrix} wr_{1,1}(n-1) & & & & ws_1(n-1) \\ & wr_{2,2}(n-1) & & & ws_2(n-1) \\ & & \ddots & & \vdots \\ & & & wr_{N,N}(n-1) & ws_N(n-1) \\ x_1(n) & x_2(n) & \cdots & x_N(n) & d(n) \end{bmatrix} \quad (7)$$

With this special structured-approximation of the triangulated augmented data matrix (c.f. step 2 in table 1), the A-QR-LS algorithm is able to combine the updating and the back solving processes together using the Householder transformation, yielding a very efficient algorithm. Liu *et al.* also proposed a related QR-LMS algorithm [5]. In [7], Chan and Yang showed that A-QR-LS is equivalent to an element-wise normalized LMS algorithm with time-varying step-sizes which will reduce

to [5] when all the normalization constants are chosen as the Euclidean norm of the input signal vector. They also proposed an improved transform domain approximate QR-LS (TA-QR-LS) algorithm based on the Givens rotation. Moreover, unitary transformation such as DCT is employed to improve the convergence speed when the input is colored.

3. THE p -TA-QR-LS ADAPTIVE ALGORITHM

3.1. Derivation of the new algorithm

From (7), we can see that the A-QR-LS algorithm retains an approximate triangular factor of the augmented data matrix so as to reduce the arithmetic complexity to $O(N)$. It is therefore natural to expect that better performance can be achieved by retaining more off diagonal elements of this factor. Obviously, when the whole upper triangular matrix is retained, we obtain the QR-RLS algorithm but the back-solving step has a complexity of $O(N^2)$. In this paper, we proposed to retain the main diagonal and $p-1$ nearby off-diagonals of the triangular factor, hence the name p -A-QR-LS algorithm. We shall show later that the p -A-QR-LS algorithm with a given positive integer p has a complexity of order $O(Np)$ and a performance, which generally improves as p increases. Therefore, the family not only provides a link between the QR-LMS-type and QR-RLS-type algorithms, but also a practical tradeoff between performance and complexity when p is varied from 1 to N .

In the proposed p -A-QR-LS algorithm, the diagonal as well as nearby $(p-1)$ off-diagonals are retained. This yields:

$$\mathbf{D}_p(n-1) = \begin{bmatrix} r_{1,1}(n-1) & r_{1,2}(n-1) & \dots & r_{1,p}(n-1) & 0 & \dots & 0 & 0 \\ & r_{2,2}(n-1) & \dots & r_{2,p}(n-1) & r_{2,p+1}(n-1) & 0 & \theta & 0 \\ & \theta & \ddots & \ddots & \ddots & \ddots & 0 & \vdots \\ & & & & & & \ddots & \vdots \\ & & & & & & & r_{Np+1,N}(n-1) \\ & & & & r_{N-2,N-2}(n-1) & r_{N-2,N-1}(n-1) & \vdots & \vdots \\ 0 & 0 & \dots & & 0 & r_{N-1,N-1}(n-1) & r_{N-1,N}(n-1) & \vdots \\ 0 & 0 & \dots & & & 0 & r_{N,N}(n-1) & \vdots \end{bmatrix} \quad (8)$$

Motivated by the A-QR-LS algorithm, (6) is now modified to

$$\begin{aligned} \Phi(n) &= \begin{bmatrix} w\mathbf{D}_p(n-1) \\ \mathbf{x}_N^T(n) \end{bmatrix} \quad \text{and} \\ \mathbf{b}(n) &= \begin{bmatrix} w\mathbf{D}_p(n-1)\theta(n-1) \\ d(n) \end{bmatrix} = \begin{bmatrix} ws(n-1) \\ d(n) \end{bmatrix}. \end{aligned} \quad (9)$$

The QRD is then applied to solve for (6) by eliminating all the elements of $\mathbf{x}_N^T(n)$ sequentially using Givens rotations. The procedure is very similar to that used in the TA-QR-LS algorithm [7]. However, the back substitution is considerably different from that in [7]. Since the

elements of both the diagonal and the $(p-1)$ off-diagonals are retained, the back substitution consists of two parts. The first part, for the nonzero diagonal and off-diagonal entries, is similar to the conventional back-substitution of the QR-RLS algorithm. It has a complexity of $O(Np)$. The second part consists of the approximation resulting from the zero off-diagonal entries, and it can be updated at a lower complexity similar to the TA-QR-LS algorithm. It has a complexity of $O(N)$. Also for further reducing the arithmetic complexity and improving the convergence speed, two techniques in [7] are adopted for the p -A-QR-LS algorithm: the SRF Givens rotation method and the transform domain preprocessing (such as DCT). The p -TA-QR-LS algorithm so obtained is summarized in Table 2. From (8) and Table 2, it can be seen that when $p=1$, the proposed algorithm reduces to the TA-QR-LS algorithm with a complexity of $O(N)$, and when $p=N$, it equals to the QR-RLS algorithm with a complexity of $O(N^2)$. With this additional flexibility, the new algorithm can be tailored for various applications with different performance and complexity requirements. It is also possible to select a p that will yield an algorithm with low complexity while the performance will remain comparable to that of the QR-RLS algorithm. Derivation of a similar complex version of the algorithm only requires little extra efforts and the details are omitted.

3.2. Mean convergence Analysis

By applying the techniques in [5, 7], the solution of equation (6) with respect to different p can be written formally as

$$\begin{aligned} \theta(n) &= (\Phi^T(n)\Phi(n))^{-1}\Phi^T(n)\mathbf{b}(n) \\ &= (w^2\mathbf{D}_p^T(n-1)\mathbf{D}_p(n-1) + \mathbf{x}_N(n)\mathbf{x}_N^T(n))^{-1} \\ &\quad \times [w\mathbf{D}_p^T(n-1) \quad \mathbf{x}_N(n)] \begin{bmatrix} w\mathbf{D}_p(n-1)\theta(n-1) \\ d(n) \end{bmatrix}. \end{aligned} \quad (10)$$

By denoting $\hat{\mathbf{D}}_p^{-1}(n) = (\mathbf{D}_p^T(n)\mathbf{D}_p(n))^{-1}$ and using the matrix inversion lemma, (10) can be simplified to

$$\begin{aligned} \theta(n) &= \left[(w^{-2}\hat{\mathbf{D}}_p^{-1}(n-1)) - \frac{(w^{-2}\hat{\mathbf{D}}_p^{-1}(n-1))(\mathbf{x}_N(n)\mathbf{x}_N^T(n))(w^{-2}\hat{\mathbf{D}}_p^{-1}(n-1))}{1 + \mathbf{x}_N^T(n)(w^{-2}\hat{\mathbf{D}}_p^{-1}(n-1))\mathbf{x}_N(n)} \right] \\ &\quad \times (w^2\hat{\mathbf{D}}_p(n-1)\theta(n-1) + d(n)\mathbf{x}_N(n)) \\ &= \theta(n-1) + \frac{\hat{\mathbf{D}}_p^{-1}(n-1)}{w^2 + (\mathbf{x}_N^T(n)\hat{\mathbf{D}}_p^{-1}(n-1)\mathbf{x}_N(n))} \mathbf{x}_N(n)(d(n) - \mathbf{x}_N^T(n)\theta(n-1)) \end{aligned} \quad (11)$$

which clearly shows that with p varying from 1 to N , $\hat{\mathbf{D}}_p(n)$ serves as an approximation of the input correlation matrix to different levels and varies from a diagonal matrix to its exact LS estimate. Assume the desired signal is given by

1. Initialization $\theta = [0, 0, \dots, 0]^T$ $\delta = [1, 1, \dots, 1]^T$, $\alpha_0 = 1$	
2. Recursive Operations $b = D_p \theta$, $\delta' = w \delta$	
Add New Equation and DCT Transform $b' = DCT(x_d)$	
<u>Upper triangular Algorithm</u> For $i = 1, 2, \dots, N$ Loop	
$\alpha' = \alpha_{i-1} \tilde{b}_{N+1,i}^{(i)}$, $\delta_i = \delta'_i + \alpha \tilde{b}_{N+1,i}^{(i)}$, $\sigma_i = \delta'_i / \delta_i$, $\rho_i = \alpha' / \delta_i$, $\alpha_i = \alpha_{i-1} \sigma_i$, $\tau = b_{i,N+1}$,	} 8N Multiplications 3N Additions
$b_{i,N+1}^* = \sigma b_{i,N+1} + \rho b_{N+1,N+1}^{(i)}$, $b_{N+1,N+1}^{(i+1)} = b_{N+1,N+1}^{(i)} - \tau b_{N+1,i}$	
If $i > N - p$ OR $j < i + p$	
$\tilde{b}_{N+1,i}^{(i+1)} = \tilde{b}_{N+1,i}^{(i)} - \tilde{b}_{i,j} \tilde{b}_{N+1,i}^{(i)}$, $\tilde{b}_{i,j}^* = \sigma \tilde{b}_{i,j} + \rho \tilde{b}_{N+1,i}^{(i)}$	} (3/2)*(2N-p)(p-1) Multiplications (2N-p)(p-1) Additions
End	
End of Loop	
<u>Back Substitution</u> $\gamma_{1-N-1} = 0$, $\tilde{\theta}(N) = b_{N,N+1}^*$	
For $i = N-1, N-2, \dots, N-p+1$ Loop	} p(p-1)/2 Multiplications p(p-1)/2 Additions
For $j = N, N-1, \dots, i+1$ Loop $\gamma_i = \gamma_j + \tilde{b}_{i,j}^* \tilde{\theta}(j)$	
End of Loop $\tilde{\theta}(i) = b_{i,N+1}^* - \gamma_i$	
End of Loop $\hat{\gamma}_{N-p+1} = 0$	
For $i = N-p, N-p-1, \dots, 2, 1$ Loop	} (N-p)(p+1) Multiplications (N-p)(p+1) Additions
$\hat{\gamma}_i = \hat{\gamma}_{i+1} + b_{N+1,i+1} \tilde{\theta}(i+p)$ $\tilde{b}_{i,N+1}^* = b_{i,N+1}^* - \rho_i \hat{\gamma}_i$	
For $j = i+p, \dots, i+1$ Loop $\gamma_i = \gamma_j + \tilde{b}_{i,j}^* \tilde{\theta}(j)$	
End of Loop $\tilde{\theta}(i) = \tilde{b}_{i,N+1}^* - \gamma_i$	
End of Loop	

Table 2. Square-root free Givens rotation-based p -TA-QR-LS algorithm.

$d(n) = \mathbf{x}_N^T(n) \theta^\circ + \eta(n)$. Substituting it into (11) and letting $V(n) = \theta(n) - \theta^\circ$ be the coefficient estimation error vector gives

$$V(n) = \left(\mathbf{I} - \frac{\hat{\mathbf{D}}_p^{-1}(n-1) \mathbf{x}_N(n) \mathbf{x}_N^T(n)}{w^2 + (\mathbf{x}_N^T(n) \hat{\mathbf{D}}_p^{-1}(n-1) \mathbf{x}_N(n))} \right) V(n-1) + \tilde{\eta}(n) \quad (12)$$

where

$$\tilde{\eta}(n) = (\hat{\mathbf{D}}_p^{-1}(n-1) \mathbf{x}_N(n) \eta(n)) / (w^2 + (\mathbf{x}_N^T(n) \hat{\mathbf{D}}_p^{-1}(n-1) \mathbf{x}_N(n)))$$

When n is sufficiently large and assume the input process is stationary, the matrix $\mathbf{D}_p(n)$ will converge to a constant matrix \mathbf{D}_p and therefore in the following analysis $\hat{\mathbf{D}}_p^{-1}(n)$ is treated as a constant matrix $\hat{\mathbf{D}}_p^{-1}$. Denoting $\mathbf{x}_N(n)$ as the output of a unitary transformation \mathbf{T} of the input vector $\boldsymbol{\varphi}(n)$, i.e., $\mathbf{T}\boldsymbol{\varphi}(n) = \mathbf{x}_N(n)$, and for simplicity, assuming \mathbf{T} is real so that $\mathbf{T}^{-1} = \mathbf{T}^T$, we can rewrite (12) as

$$V(n) = \left\{ \mathbf{I} - \frac{\hat{\mathbf{D}}_p^{-1} \mathbf{T} \boldsymbol{\varphi}(n) \boldsymbol{\varphi}^T(n) \mathbf{T}^T}{w^2 + (\boldsymbol{\varphi}^T(n) \mathbf{T}^T \hat{\mathbf{D}}_p^{-1} \mathbf{T} \boldsymbol{\varphi}(n))} \right\} \cdot V(n-1) + \tilde{\eta}(n) \quad (13)$$

By defining $\mathbf{Y}(n) = \mathbf{D}_p^{-1} \mathbf{T} \cdot \boldsymbol{\varphi}(n)$ and substituting it into (13),

$$\text{we get } V'(n) = \left\{ \mathbf{I} - \frac{\mathbf{Y}(n) \mathbf{Y}^T(n)}{w^2 + \mathbf{Y}^T(n) \mathbf{Y}(n)} \right\} \cdot V'(n-1) + \tilde{\eta}(n) \quad (14)$$

where $V'(n) = \mathbf{D}_p V(n)$. Let us evoke the classic independent assumption of $\boldsymbol{\varphi}(n)$ (thus $\mathbf{Y}(n)$ is also independent), and further assume that the estimation error vector $V(n)$ is uncorrelated with the transformed input signal vector $\mathbf{Y}(n)$ when the stepsize is sufficiently small. Then, taking expectation on both sides of (14) and noticing that $E[\tilde{\eta}(n)] = \theta$ yields

$$E[V'(n)] = \left\{ \mathbf{I} - E \left[\frac{\mathbf{Y}(n) \mathbf{Y}^T(n)}{w^2 + \mathbf{Y}^T(n) \mathbf{Y}(n)} \right] \right\} E[V'(n-1)] \\ = \{ \mathbf{I} - \mathbf{A} \}^n E[V'(0)] \quad (15)$$

where $\mathbf{A} = E[(\mathbf{Y}(n) \mathbf{Y}^T(n)) / (w^2 + \mathbf{Y}^T(n) \mathbf{Y}(n))]$.

Consider the trace (tr) of the matrix \mathbf{A} , following the linearity of the trace and expectation operators and the

$$\text{fact that } 0 < w < 1, \text{ we have } tr(\mathbf{A}) = tr \left(E \left[\frac{\mathbf{Y}(n) \mathbf{Y}^T(n)}{w^2 + \mathbf{Y}^T(n) \mathbf{Y}(n)} \right] \right) \\ = E \left[\frac{tr(\mathbf{Y}(n) \mathbf{Y}^T(n))}{w^2 + \mathbf{Y}^T(n) \mathbf{Y}(n)} \right] = E \left[\frac{\mathbf{Y}^T(n) \mathbf{Y}(n)}{w^2 + \mathbf{Y}^T(n) \mathbf{Y}(n)} \right] < 1 \quad (16)$$

Since the eigenvalues $\lambda_i, i = 1, \dots, N$ of the symmetric matrix \mathbf{A} are all positive and $tr(\mathbf{A}) = \sum_{i=1}^N \lambda_i$, it follows that $0 < \lambda_i \leq tr(\mathbf{A}) < 1$, and therefore, the mean of the estimation error vector in (15) is guaranteed to converge to a steady-state value $E[V'(\infty)] = 0$.

4. SIMULATION RESULTS

We now evaluate the performance of the proposed p -A-QR-LS and p -TA-QR-LS algorithms using computer simulation. The algorithm with different parameter p is tested in an adaptive system coefficient estimation problem. The order of the unknown system is set to 14 and the coefficients are randomly generated. The power of the additive white Gaussian noise $\eta(n)$ is set to be 0.0001. The forgetting factor w for all the tested algorithms are equally set to be 0.99. The Mean Square Difference (MSD), defined as the sum of the square of the coefficient estimation errors, is used to evaluate the convergence behavior of the algorithms. All the results are averaged over 100 independent runs. Three experiments were

conducted. **Exp.1:** input is a white Gaussian noise (WGN) sequence with zero mean and unit variance, no transformation employed; **Exp.2:** input is a WGN-driven AR process whose parameters are fixed at $[1 \ -1.5 \ 1 \ -0.25]$ with normalized unit power, no transformation employed; **Exp. 3:** input is the same as in **Exp.2**, DCT is employed. Fig. 1 shows the results of **Exp.1**. All the algorithms with different p approach the same steady state error, whereas when p increases, the algorithm with larger p has faster initial convergence, which complies well with our previous analyses. Fig. 2 shows the results of **Exp.2** and a similar conclusion can be drawn except that the convergence speed of those algorithms with small p is hindered by the colored input. However, when the DCT is included, the results of **Exp.3** as depicted in Fig. 3 show that the convergence speed for the p -TA-QR-LS algorithms with small p is significantly improved. All these observations verify the flexibility of the proposed algorithm and reveal its potential to be applied to different applications.

5. CONCLUSION

A new family of approximate QR-based LS adaptive algorithms called p -TA-QR-LS algorithms is presented. It retains different number of diagonal plus off-diagonals in triangular factor of the augmented data matrix and generates a series of new QR-based algorithms with complexity-performance tradeoff between the QR-LMS-type and QR-RLS-type algorithms. These results are verified by computer simulation and the mean convergence of the algorithms is also analyzed.

6. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*. 4th edition, Prentice Hall Press, 2001.

[2] J. G. Proakis, C. M. Rader, F. Ling, C. L. Nikias, M. Moonen, and I. K. Proudler. *Algorithms for Statistical signal Processing*. Prentice Hall Press, New Jersey, 2002.

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

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

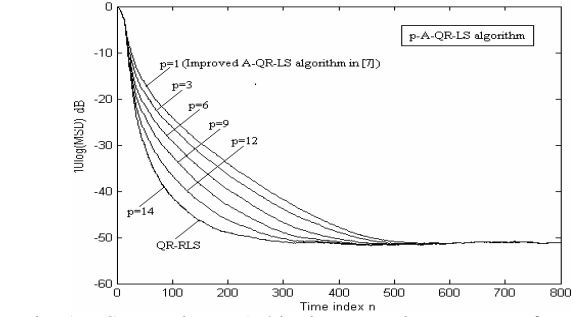


Fig. 1 MSD vs. time n (white input, $w=0.99$, no transformation employed)

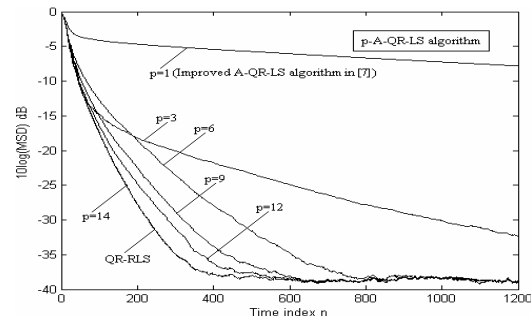


Fig. 2 MSD vs. time n (colored input, $w=0.99$, no transformation employed)

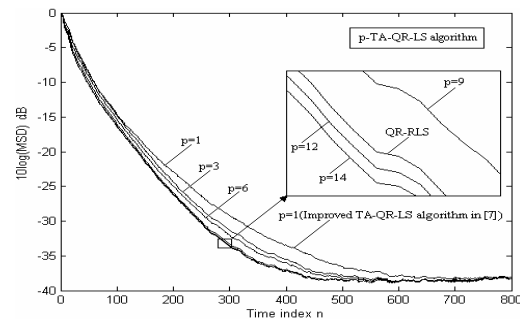


Fig. 3 MSD vs. time n (colored input, $w=0.99$, DCT employed)

[5] M. Bhourri, M. Bonnet, and M. Mboup, "A new QRD-based block adaptive algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1998, pp. 1497-1500.

[6] S. C. Chan and X. X. Yang, "Improved Approximate QR-LS Algorithms for Adaptive Filtering," *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 51, No. 1, pp. 29-39, Jan. 2004.

[7] X. X. Yang and S. C. Chan, "A new QR-based block least mean squares (QR-BLMS) algorithm for adaptive parameter estimation," *ICASSP'2003*, vol. 6, pp. 653-656, Apr. 2003.

[8] G. V. Moustakides and S. Theodoridis, "Fast newton transversal filters-a new class of adaptive estimation algorithms," *IEEE Trans. on Signal Processing*, vol. 39, No. 10, pp. 2184-2193, Oct. 1991.