

Improved Approximate QR-LS Algorithms for Adaptive Filtering

S. C. Chan and X. X. Yang

Abstract—This paper studies a class of $O(N)$ approximate QR-based least squares (A-QR-LS) algorithm recently proposed by Liu in 1995. It is shown that the A-QR-LS algorithm is equivalent to a normalized LMS algorithm with time-varying stepsizes and element-wise normalization of the input signal vector. It reduces to the QR-LMS algorithm proposed by Liu *et al.* in 1998, when all the normalization constants are chosen as the Euclidean norm of the input signal vector. An improved transform-domain approximate QR-LS (TA-QR-LS) algorithm, where the input signal vector is first approximately decorrelated by some unitary transformations before the normalization, is proposed to improve its convergence for highly correlated signals. The mean weight vectors of the algorithms are shown to converge to the optimal Wiener solution if the weighting factor w of the algorithm is chosen between 0 and 1. New Givens rotations-based algorithms for the A-QR-LS, TA-QR-LS, and the QR-LMS algorithms are proposed to reduce their arithmetic complexities. This reduces the arithmetic complexity by a factor of 2, and allows square root-free versions of the algorithms to be developed. The performances of the various algorithms are evaluated through computer simulation of a system identification problem and an acoustic echo canceller.

Index Terms—Adaptive filtering, approximate QR-LS algorithm, performance analysis, QR-LMS algorithm, square root free Givens based algorithms, transformed domain LMS algorithm.

I. INTRODUCTION

ADAPTIVE filters have been widely used in communications, control, and many other systems in which the statistical characteristics of the signals to be filtered are either unknown *a priori* or, in some cases, slowly time varying. Two commonly used families of adaptive filtering algorithms are based on the least mean squares (LMS) and the recursive least squares (RLS) algorithms [15]–[19]. The RLS-based adaptive algorithms are well known for their fast convergence speed, as compared with the LMS-based algorithms. The convergence speed of the latter is usually very sensitive to the eigenvalue spread of the correlation matrix of the input signal. The LMS-based algorithms, however, has a very low arithmetic complexity of $O(N)$ (where N is the number of taps in the adaptive filter), as compared with $O(N^2)$ for the conventional RLS algorithm. Another problem with the classical RLS algorithm is its numerical instability usually encountered in finite wordlength implementation, especially for short internal wordlength. These problems have stimulated

extensive research into efficient methods for reducing the arithmetic complexity and improving the numerical properties of the RLS algorithms [1]–[11], [20]–[24], [37]–[39]. Fast RLS algorithms can broadly be classified into two different categories. The first class of algorithm recursively updates the inverse of the correlation matrix of the input signal using certain time- and order-recursions [1]–[5], [9]–[11], [21], [22], [39]. The second class of algorithms works directly with the data matrix using QR decomposition (QRD) [Givens rotation or Householder transformation] [6]–[8], [20]–[24], [37], [38], [40]–[42]. Fast RLS algorithms using the QRD usually exhibit better numerical property because of their lower condition number of the system, as compared with that of the input correlation matrix, which is square of that of the data matrix. Based on these two approaches, several fast RLS adaptive filtering algorithms with $O(N)$ complexity are now available [1]–[5], [8], [11], [18], [23], [24], [39]. Interested readers are referred to the excellent summary in [15, pp. 394, 469–470] and a recent survey paper in [18]. Algorithms employing the QRD usually consist of the following two separate parts: 1) recursive updating of the triangular matrix and 2) backsolving of the parameters. Since the backsolving step requires $O(N^2)$ operation, the entire algorithm still requires at least $O(N^2)$ arithmetic operations. For single input adaptive filters, the time-shift property of the input signal can further be exploited to reduce the arithmetic complexity for updating the triangular factor in $O(N)$ complexity. This is useful in beamforming and some applications, where only the least squares error is of interest. For multichannel adaptive filtering [8], [21]–[24], however, the input signal vector does not satisfy the time-shifting property and a higher complexity is required. In order to reduce the complexity of the back-substitution step in the QRD using Householder transformation, Liu [12] proposed an approximate QR-LS algorithm, which combines the recursive updating of the triangular matrix and the back-solving of the parameters. A related QR-LMS algorithm based on Householder transformation was also proposed recently in [13]. It was shown that the QR-LMS algorithm is mathematically equivalent to the ONM-LMS algorithm [35] in infinite precision arithmetic. However, the QR-LMS algorithm generally exhibits better numerical property than the ONM-LMS algorithm because of its lower conditional number in solving for the parameter vectors. This benefit, however, is realized at the cost of more computations ($17N$ multiplications, $9N$ additions, and N square root), though both algorithms have $O(N)$ complexity.

In this paper, we show that the approximate QR-based LS (A-QR-LS) algorithm in [12] is in fact equivalent to an element-wise normalized LMS algorithm with time-varying step-

Manuscript received August 8, 2002; revised July 30, 2003. This paper was recommended by Associate Editor S.-M. Phoong.

S. C. Chan is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: scchan@eee.hku.hk).

X. X. Yang is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: xxyang@eee.hku.hk).

Digital Object Identifier 10.1109/TCSII.2003.821514

sizes. It reduces to the QR-LMS algorithm when all the normalization constants are chosen as the Euclidean norm of the input signal vector (as pointed out by one of the reviewer, it is possible to come up with a similar conclusion using the results reported in a recent work by Bhourri *et al.* [42], which deals with the block version of this algorithm). Although the normalization of individual elements in the input signal vector improves the convergence speed by reducing the eigenvalue spread of the input correlation matrix, such improvement is limited when the components are highly correlated. An improved transform-domain approximate QR-LS (TA-QR-LS) algorithm, where the input signal vector is first approximately decorrelated by some unitary transformations such as the discrete cosine transform (DCT) before carrying out the normalization is, therefore, proposed. This considerably improves the convergence speed of the A-QR-LS algorithm. The mean convergence behaviors of the A-QR-LS and TA-QR-LS algorithms are analyzed and it suggests that the mean weight of the algorithms converge to the optimal Wiener solution if the weighting factor w of the algorithms is chosen between 0 and 1. The TA-QR-LS algorithm can be viewed as a variable stepsize and numerical better-behaved implementation of the transform domain LMS (TLMS) algorithm [34]. An improved TA-QR-LS with initial power estimation, called the power estimated transform domain approximate QR-LS (PTA-QR-LS), is also proposed. Further, we show that it is possible to reduce the arithmetic complexities of the A-QR-LS, TA-QR-LS, PTA-QR-LS, and the QR-LMS algorithms by using Givens rotations instead of the Householder transformation, as only one input signal vector is processed at a time. This reduces the arithmetic complexity by a factor of two, and facilitates the development of new square root-free versions of the algorithms, in the same spirit as the classical square root-free Givens-based QR decomposition LS algorithm [6]. The resultant algorithms are much simpler to implement in either software or hardware. The performances of the various algorithms are evaluated through computer simulation. Simulation results show that the proposed algorithms are good alternatives to the RLS and QR-LMS algorithms in applications involving multiple channels, acoustic modeling, and fast parameter variations. The rest of the paper is organized as follows. The traditional QR-based LS adaptive algorithm is briefly reviewed in Section II. The approximate QR-LS algorithm proposed in [13] and the new TA-QR-LS algorithm are described in Section III. The mean convergence performance analysis and its relation to the normalized LMS algorithm will also be given. Section IV is devoted to the efficient implementation of the various approximate QR-LS algorithms. Simulation results and comparison of the various algorithms are carried out in Section V. Finally, conclusions are drawn in Section VI.

II. QR-BASED LS ADAPTIVE ALGORITHM

Consider the estimation of the N -dimensional parameter vector θ° for the following linear model:

$$d(n) = \mathbf{x}_N^T(n)\theta^\circ + v(n) \quad (1)$$

where $d(n)$ and $\mathbf{x}_N^T(n) = [x_1(n), x_2(n), \dots, x_N(n)]$ are the desired (observed) signal and input vectors, respectively, and $v(n)$ is an additive white Gaussian noise sequence with zero

mean. Let $\theta(n)$ be the estimated parameter vector at time n , the estimation error at time instant n is, thus, given by

$$e(j) = d(j) - \mathbf{x}_N^T(j)\theta(n). \quad (2)$$

In least squares parameter estimation, the following time-averaged squared magnitude error $\xi_N(n)$ is:

$$\begin{aligned} \xi_N(n) &= \sum_{j=0}^n \lambda^{n-j} |e(j)|^2 \\ &= \sum_{j=0}^n \lambda^{n-j} |d(j) - \mathbf{x}_N^T(j)\theta(n)|^2 \end{aligned} \quad (3)$$

where the constant λ is the forgetting factor with a value between 0 and 1. Equation (1) can be written more compactly in matrix form as

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}_N(n)\theta(n) \quad (4)$$

where

$$\mathbf{d}(n) = [d(0), d(1), \dots, d(n)]^T$$

$$\mathbf{x}_N(n) = [x_1(n), x_2(n), \dots, x_N(n)]^T \quad (5)$$

$$\mathbf{X}_N(n) = [\mathbf{x}_N(0), \mathbf{x}_N(1), \dots, \mathbf{x}_N(n)]^T. \quad (6)$$

$\mathbf{x}_N(n)$ and $\mathbf{X}_N(n)$ are the received signal vector and the data matrix, respectively. Then, the least squares objective function $\xi_N(n)$ in (3) becomes

$$\xi_N(n) = \mathbf{e}^H(n)\mathbf{W}^2(n)\mathbf{e}(n) = \|\mathbf{W}(n)\mathbf{e}(n)\|^2 \quad (7)$$

where $\mathbf{W}(n)$ is a diagonal weighting matrix given by $\mathbf{W}(n) = \text{diag}(\sqrt{\lambda^n}, \sqrt{\lambda^{n-1}}, \dots, \sqrt{\lambda}, 1)$. The optimum value of $\theta(n)$ can be obtained by solving the normal equation: $\mathbf{R}_N(n)\theta(n) = \mathbf{p}_N(n)$, where $\mathbf{R}_N(n) = \sum_{i=0}^n \lambda^{n-i} \mathbf{x}_N(i)\mathbf{x}_N^H(i)$ and $\mathbf{p}_N(n) = \sum_{i=0}^n \lambda^{n-i} d(i)\mathbf{x}_N^H(i)$ are, respectively, the weighted autocorrelation matrix of $\mathbf{x}_N(n)$, and the weighted cross-correlation vector of $\mathbf{x}_N(n)$ and $\mathbf{d}(n)$. Due to the lower numerical accuracy in solving the normal equation, a better method, called the QR-LS method, is employed. The following QRD of $\mathbf{W}(n)\mathbf{X}_M(n)$ is performed

$$\mathbf{Q}(n)\mathbf{W}(n)\mathbf{X}_N(n) = \begin{bmatrix} \mathfrak{R}_N(n) \\ \mathbf{0} \end{bmatrix} \quad (8)$$

where $\mathbf{Q}(n)$ is some $(n+1) \times (n+1)$ unitary matrix and $\mathfrak{R}_N(n)$ is an $(N \times N)$ upper triangular matrix. Using (8), (4) can be rewritten as

$$\begin{aligned} \mathbf{Q}(n)\mathbf{W}(n)\mathbf{e}(n) &= \begin{bmatrix} \hat{\mathbf{d}}_N(n) \\ \mathbf{c}_{n+1-N} \end{bmatrix} - \begin{bmatrix} \mathfrak{R}_N(n) \\ \mathbf{0} \end{bmatrix} \theta(n) \\ &= \begin{bmatrix} \hat{\mathbf{d}}_N(n) - \mathfrak{R}_N(n)\theta(n) \\ \mathbf{c}_{n+1-N} \end{bmatrix} \end{aligned} \quad (9)$$

where $\mathbf{Q}(n)\mathbf{W}(n)\mathbf{d}(n) = [\hat{\mathbf{d}}_N(n) \ \mathbf{c}_{n+1-N}]^T$. Since $\mathbf{Q}(n)$ is an unitary matrix, the square of the Euclidean norm on the left-hand side in (9) is equal to $\xi_N(n)$ in (7). The two-norm on the right-hand side of (9) achieves its minimum value when $\theta(n)$ is chosen as $\mathfrak{R}_N(n)\theta(n) = \hat{\mathbf{d}}_N(n)$, and $\min_{\theta(n)} \xi(n) = \xi_N^*(n) = \|\mathbf{c}_{n+1-N}\|_2^2$. Since $\mathfrak{R}_N(n)$ is an upper triangular matrix, $\theta(n)$ can be obtained by back-substitution. There are several methods to perform the QRD of the weighted data matrix $\mathbf{W}(n)\mathbf{X}_N(n)$. If a single input vector $\mathbf{x}_N(n)$ is processed at a time, then the Givens rotation-based

QRD algorithm is preferred because of its low arithmetic complexity. When multiple input vectors $\mathbf{x}_N(n)$'s are processed at a time, then the Householder transformation [15] is more efficient. The QR-LS algorithm can be made recursive for each input vector. Let's define the augmented data matrix

$$\mathbf{D}_N(n-1) = \mathbf{W}(n-1)[\mathbf{X}_N(n-1) \quad \mathbf{d}_N(n-1)]. \quad (10)$$

Suppose that we have computed the QRD of $\mathbf{W}(n-1)\mathbf{X}_N(n-1)$, then we have

$$\begin{aligned} \mathbf{D}_N^*(n-1) &= \mathbf{Q}(n-1)\mathbf{D}_N(n-1) \\ &= \begin{bmatrix} \mathfrak{R}_N(n-1) & \hat{\mathbf{d}}_N(n-1) \\ \mathbf{0} & \mathbf{c}_{n-N} \end{bmatrix}. \end{aligned} \quad (11)$$

Given the new data vector $\psi_{N+1}(n) = [\mathbf{x}_N^T(n), d(n)]^T$, we are interested in computing the QRD of $\mathbf{W}(n)\mathbf{X}_N(n)$. From (10), we have

$$\mathbf{D}_N(n) = \mathbf{W}(n)[\mathbf{X}_N(n) \quad \mathbf{d}(n)] = \begin{bmatrix} \sqrt{\lambda}\mathbf{D}_N(n-1) \\ \psi_{N+1}^T(n) \end{bmatrix}. \quad (12)$$

Multiply (12) by the augmented matrix

$$\mathbf{Q}'(n) = \begin{bmatrix} \mathbf{Q}(n-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

one gets

$$\begin{aligned} \mathbf{Q}'(n)\mathbf{D}_N(n) &= \begin{bmatrix} \sqrt{\lambda}\mathbf{Q}(n-1)\mathbf{D}_N(n-1) \\ \psi_{N+1}^T(n) \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{\lambda}\mathfrak{R}_N(n-1) & \sqrt{\lambda}\hat{\mathbf{d}}_N(n-1) \\ \mathbf{0} & \sqrt{\lambda}\mathbf{c}_{n-N} \\ \mathbf{x}_N^T(n) & d(n) \end{bmatrix}. \end{aligned} \quad (13)$$

Since $\mathfrak{R}_N(n-1)$ is an upper-triangular matrix, the new QRD can be obtained by zeroing out $\mathbf{x}_N^T(n)$ by a series of N -Givens rotations or Householder reflections

$$\begin{aligned} \mathbf{Q}^{(N)}(n) \cdots \mathbf{Q}^{(1)}(n)\mathbf{Q}'(n)\mathbf{D}_N(n) \\ = \begin{bmatrix} \mathfrak{R}_N(n) & \hat{\mathbf{d}}_N(n) \\ \mathbf{0} & \mathbf{c}_{n-N} \\ \mathbf{0}^T & d^{N+1}(n) \end{bmatrix} \end{aligned} \quad (14)$$

where $\mathbf{Q}^{(i)}(n)$ are the Givens rotation or Householder matrix used to zero out the element $x_i(n)$ at the i th stage. Normally, the matrix $\mathbf{Q}(n)$ is not stored explicitly. Only the triangular matrix $\mathfrak{R}_N(n)$ is stored and the minimum LS error $\xi_N^*(n)$ is recursively computed as $\xi_N^*(n) = \xi_N^*(n-1) + (d^{N+1}(n))^2$.

III. THE TRANSFORM-DOMAIN APPROXIMATE QR-LS ALGORITHM

Even though it is possible to recursively update the triangular matrix for single-input adaptive-filtering applications with $O(N)$ arithmetic operations, direct back solving of the optimal parameter vector $\boldsymbol{\theta}^o$ still requires about $O(N^2)$ arithmetic complexity. In [12], Liu proposed an approximate QR-LS algorithm, which approximates the triangular factor with a special structural matrix. By so doing, it is possible to combine the updating and the back solving processes together using the Householder transformation, yielding a very efficient algorithm requiring N square roots, $17N$ multiplications, and $9N$ additions. However, the performance analysis of this algorithm was not studied. Here we show that this approximate QR-LS algorithm

is closely related to the normalized LMS algorithm and analyze its mean convergence performance. In addition, a new transform domain version of this approximate QR-LS algorithm is proposed, which has a faster convergence speed than the original approximate QR-LS algorithm when the input correlation matrix is highly correlated. First of all, let us briefly summarize the principle of this approximate QR-LS algorithm. Assume that the upper triangular matrix $\mathfrak{R}_N(n-1)$ is known at time instant $n-1$. From $\mathfrak{R}_N(n)\boldsymbol{\theta}(n) = \hat{\mathbf{d}}_N(n)$ in (9), the parameter vector can be computed by back substitution as follows:

$$\begin{aligned} \theta_N(n-1) &= -[r_{N,N+1}(n-1)]/r_{N,N}(n-1) \\ \theta_i(n-1) &= - \left[r_{i,N+1}(n-1) \right. \\ &\quad \left. + \sum_{j=i+1}^N r_{i,j}(n-1)\theta_j(n-1) \right] / r_{i,i}(n-1) \\ &\quad i = N-1, \dots, 1 \end{aligned} \quad (15)$$

where $\theta_i(n-1)$ is the i th element of the estimated parameter vector $\boldsymbol{\theta}(n-1)$. Let the term inside the square bracket for $\theta_i(n-1)$ in (15) be $s_i(n-1)$. Then

$$\begin{aligned} s_N(n-1) &= -r_{N,N+1}(n-1) \\ s_i(n-1) \\ &= - \left[r_{i,N+1}(n-1) + \sum_{j=i+1}^N r_{i,j}(n-1)\theta_j(n-1) \right] \\ &\quad i = N-1, N-2, \dots, 1 \end{aligned} \quad (16)$$

and (15) can be rewritten as

$$r_{i,i}(n-1)\theta_i(n-1) = s_i(n-1), \quad i = 1, \dots, N. \quad (17)$$

Given the values of $s_i(n-1)$ and $r_{i,i}(n-1)$, (17) can be viewed as a system of linear equations in variable $\boldsymbol{\theta}(n)$, the parameter estimate to be solved at time instant n . This, together with (1), yields the following equation in $\boldsymbol{\theta}(n)$:

$$\begin{aligned} w \cdot r_{i,i}(n-1)\theta_i(n) &= w \cdot s_i(n-1), \quad i = 1, \dots, N \\ \mathbf{x}_N^T(n)\boldsymbol{\theta}(n) &= d(n) \end{aligned} \quad (18)$$

where w is the square root of the forgetting factor. Note, because of the approximation just mentioned for (17), $r_{i,i}(n-1)$'s in (18) will be different from those of the QRD. To solve (18), let us rewrite it in matrix form as follows:

$$\boldsymbol{\Phi}(n)\boldsymbol{\theta}(n) = \mathbf{b}(n) \quad (19)$$

where

$$\begin{aligned} \boldsymbol{\Phi}(n) &= \begin{bmatrix} w\mathbf{D}(n-1) \\ \mathbf{x}_N^T(n) \end{bmatrix} \\ \mathbf{b}(n) &= \begin{bmatrix} w\mathbf{D}(n-1)\boldsymbol{\theta}(n-1) \\ d(n) \end{bmatrix} \\ \mathbf{D}(n-1) &= \text{diag}\{r_{1,1}(n-1), \dots, r_{N,N}(n-1)\}. \end{aligned}$$

Equation (19) can be solved by computing the QRD of $\boldsymbol{\Phi}(n)$. However, because the matrix $\mathbf{D}(n-1)$ in $\boldsymbol{\Phi}(n)$ is a diagonal matrix, the system can be solved using the QRD in order $O(N)$ arithmetic complexity. More precisely, we can construct the appended matrix $\tilde{\mathbf{D}}_N(n) = [\boldsymbol{\Phi}(n), \mathbf{b}(n)]$ as follows [12] [as seen

in (20) at the bottom of the page]. Because of the special structure of this matrix, the upper triangularization and the back solving processes can be combined together using the Householder transformation. Due to page limitation, interested readers are referred to [12] for more details of the algorithm. Because of the similarity between the QRD-based LS method and the QR method for solving (19), this is called an approximate $O(N)$ QR-LS method. However, due to the relation used in (17), it will be shown in the following that it is a variable stepsize LMS algorithm with element-wise normalization of the input signal vector.

A. The Relation With the Normalized LMS Algorithm

From (19), we know that the parameter estimate of the approximate QR-LS method is the solution of the equation $\Phi(n)\theta(n) = \mathbf{b}(n)$, which can be written formally as

$$\begin{aligned} \hat{\theta}(n) &= (\Phi^T(n)\Phi(n))^{-1}\Phi^T(n)\mathbf{b}(n) \\ &= \left\{ [w\mathbf{D}(n) \quad \mathbf{x}_N(n)] \begin{bmatrix} w\mathbf{D}(n-1) \\ \mathbf{x}_N^T(n) \end{bmatrix} \right\}^{-1} \\ &\quad \times [w\mathbf{D}(n-1) \quad \mathbf{x}_N(n)] \begin{bmatrix} w\mathbf{D}(n-1)\theta(n-1) \\ d(n) \end{bmatrix} \\ &= (w\mathbf{D}^2(n-1) + \mathbf{x}_N(n)\mathbf{x}_N^T(n))^{-1} \\ &\quad \times [w\mathbf{D}(n-1) \quad \mathbf{x}_N(n)] \begin{bmatrix} w\mathbf{D}(n-1)\theta(n-1) \\ d(n) \end{bmatrix} \end{aligned} \quad (21)$$

Using the matrix inversion lemma, (21) can be simplified to (22), as shown at the bottom of the page.

It can be seen that the recursion in (22) is very similar to the normalized LMS algorithm, except that the normalization is performed on each of the component of $\mathbf{x}_N(n)$ and a variable stepsize parameter $\mu_{\text{vss}} = [w^2 + \mathbf{x}_N^T(n)\mathbf{D}^{-2}(n-1)\mathbf{x}_N(n)]^{-1}$ is employed. From (20), we notice that when n is large, the elements of $\mathbf{D}^2(n)$ will approach the energy of the corresponding elements in $\mathbf{x}_N(n)$. Further, this normalized LMS algorithm reduces to the QR-LMS algorithm if $r_{i,i}, i = 1, \dots, N$, are equal to one. As the normalization is performed on the original input data vector, considerable improvement in the convergence speed can only be achieved when they are uncorrelated and with different signal powers. An efficient method to realize this advantage is to transform the input signal vector using certain uni-

tary or orthogonal matrices so as to approximately diagonalize the covariance matrix \mathbf{R}_N . The resulting transform domain approximate QR-LS (TA-QR-LS) algorithm becomes a variable stepsize implementation of the transform domain LMS algorithm, using the numerically well-behaved Givens rotation or Householder reflection. In Section III-B, we shall show that the condition number in solving the normal equation is improved when this QR-based algorithm is employed. Further, it is shown in Section IV that the arithmetic complexity of this algorithm can be further reduced using the Givens rotation instead of the Householder transformation. Next, we consider the mean convergence of this TA-QR-LS algorithm.

B. Performance Analysis

Assume that the desired signal is given by

$$d(n) = \mathbf{x}_N^T(n)\theta^\circ + \eta(n) \quad (23)$$

where θ° is the Wiener solution and $\eta(n)$ is a zero mean noise process uncorrelated with $\mathbf{x}_N(n)$. Substituting (23) into (22) and letting $\mathbf{V}(n) = \theta(n) - \theta^\circ$ be the error weight vector, we get

$$\mathbf{V}(n) = \left(\mathbf{I} - \frac{\mathbf{D}^{-2}(n-1)\mathbf{x}_N(n)\mathbf{x}_N^T(n)}{w^2 + (\mathbf{x}_N^T(n)\mathbf{D}^{-2}(n-1)\mathbf{x}_N(n))} \right) \cdot \mathbf{V}(n-1) + \tilde{\eta}(n) \quad (24)$$

where $\tilde{\eta}(n) = (\mathbf{D}^{-2}(n-1)\mathbf{x}_N(n)\eta(n))/(w^2 + (\mathbf{x}_N^T(n)\mathbf{D}^{-2}(n-1)\mathbf{x}_N(n)))$. If n is sufficient large, the matrix $\mathbf{D}(n)$ will converge to a constant matrix \mathbf{D} . Therefore, for simplicity, we assume the matrix $\mathbf{D}(n)$ converges to the matrix \mathbf{D} in the following analysis. For transform domain LMS algorithm, $\mathbf{x}_N(n)$ is obtained by transforming the input vector $\phi(n)$ by some unitary transformation matrix \mathbf{C} as follows

$$\mathbf{C}\phi(n) = \mathbf{x}_N(n) \quad (25)$$

where $\mathbf{C}^{-1} = \mathbf{C}^H$. For simplicity, we assume that \mathbf{C} is real and hence it is orthogonal. Then, (24) becomes

$$\mathbf{V}(n) = \left\{ \mathbf{I} - \frac{\mathbf{D}^{-2}\mathbf{C}\phi(n)\phi^T(n)\mathbf{C}^T}{w^2 + (\phi^T(n)\mathbf{C}^T\mathbf{D}^{-2}\mathbf{C}\phi(n))} \right\} \cdot \mathbf{V}(n-1) + \tilde{\eta}(n). \quad (26)$$

$$\tilde{\mathbf{D}}_N(n) = \begin{bmatrix} wr_{1,1}(n-1) & & & -ws_1(n-1) \\ & wr_{2,2}(n-1) & & -ws_2(n-1) \\ & & \mathbf{O} & \vdots \\ \mathbf{O} & & \ddots & \\ & & & wr_{N,N}(n-1) & -ws_N(n-1) \\ x_1(n) & x_2(n) & \cdots & x_N(n) & d(n) \end{bmatrix}. \quad (20)$$

$$\begin{aligned} \theta(n) &= \left[(w^{-2}\mathbf{D}^{-2}(n-1)) - \frac{(w^{-2}\mathbf{D}^{-2}(n-1))(\mathbf{x}_N(n)\mathbf{x}_N^T(n))(w^{-2}\mathbf{D}^{-2}(n-1))}{1 + \mathbf{x}_N^T(n)(w^{-2}\mathbf{D}^{-2}(n-1)\mathbf{x}_N(n))} \right] \\ &\quad \times (w^2\mathbf{D}^2(n-1)\theta(n-1) + d(n)\mathbf{x}_N(n)) = \theta(n-1) + \frac{\mathbf{D}^{-2}(n-1)}{w^2 + (\mathbf{x}_N^T(n)\mathbf{D}^{-2}(n-1)\mathbf{x}_N(n))} \\ &\quad \times \mathbf{x}_N(n)(d(n) - \mathbf{x}_N^T(n)\theta(n-1)). \end{aligned} \quad (22)$$

Substituting the transformation $\mathbf{Y}(n) = \mathbf{D}^{-1}\mathbf{C} \cdot \boldsymbol{\phi}(n)$ into (26) gives

$$\mathbf{V}'(n) = \left\{ \mathbf{I} - \frac{\mathbf{Y}(n)\mathbf{Y}^T(n)}{w^2 + \mathbf{Y}^T(n)\mathbf{Y}(n)} \right\} \cdot \mathbf{V}'(n-1) + \tilde{\eta}(n) \quad (27)$$

where $\mathbf{V}'(n) = \mathbf{D} \cdot \mathbf{V}(n)$. If we assume that $\boldsymbol{\phi}(n)$'s are uncorrelated over time indices n , then so is $\mathbf{Y}(n)$. For a sufficiently small stepsize, we can further assume that the error weight vector $\mathbf{V}(n)$ is uncorrelated with the transformed input signal vector $\mathbf{Y}(n)$. Taking expectation on both sides of (27), and using the fact that $\eta(n)$ is zero mean and uncorrelated with $\mathbf{x}_N(n)$ (i.e., $E[\tilde{\eta}(n)] = \mathbf{0}$) yields

$$E[\mathbf{V}'(n)] = \left\{ \mathbf{I} - E \left[\frac{\mathbf{Y}(n)\mathbf{Y}^T(n)}{w^2 + \mathbf{Y}^T(n)\mathbf{Y}(n)} \right] \right\} \times E[\mathbf{V}'(n-1)] = \{\mathbf{I} - \boldsymbol{\Lambda}\}^n E[\mathbf{V}'(0)] \quad (28)$$

where $\boldsymbol{\Lambda} = E[(\mathbf{Y}(n)\mathbf{Y}^T(n))/(w^2 + \mathbf{Y}^T(n)\mathbf{Y}(n))]$. From (28), it can be seen that the algorithm is convergent with a steady-state solution of $\mathbf{V}'(\infty) = \mathbf{0}$, if all the eigenvalues of the matrix $\boldsymbol{\Lambda}$ are less than 1. Consider the trace of this matrix, we have

$$\begin{aligned} \text{tr}(\boldsymbol{\Lambda}) &= \text{tr} \left(E \left[\frac{\mathbf{Y}(n)\mathbf{Y}^T(n)}{w^2 + \mathbf{Y}^T(n)\mathbf{Y}(n)} \right] \right) \\ &= E \left[\frac{\text{tr}(\mathbf{Y}(n)\mathbf{Y}^T(n))}{w^2 + \mathbf{Y}^T(n)\mathbf{Y}(n)} \right] \\ &= E \left[\frac{\mathbf{Y}^T(n)\mathbf{Y}(n)}{w^2 + \mathbf{Y}^T(n)\mathbf{Y}(n)} \right] < 1. \end{aligned} \quad (29)$$

Equation (29) follows from the linearity of the trace and expectation operators and the fact that $0 < w < 1$. Since the eigenvalues $\lambda_i, i = 1, \dots, N$ of the symmetric matrix $\boldsymbol{\Lambda}$ are all positive and $\text{tr}(\boldsymbol{\Lambda}) = \sum_{i=1}^N \lambda_i$, it follows that $0 < \lambda_i \leq \text{tr}(\boldsymbol{\Lambda}) < 1$, and the mean weight of the algorithm converges to the optimal Wiener solution. In the following, we shall briefly analyze the condition number of the QR-based method in solving the linear system in (19). First of all, we note from (19) that

$$\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n) = w\mathbf{D}^2(n-1) + \mathbf{x}_N(n)\mathbf{x}_N^T(n). \quad (30)$$

The condition number for the QR-based solution is $\text{cond}(\boldsymbol{\Phi}(n)) = \sqrt{\text{cond}(\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n))}$, and it is expected to be much smaller than solving the system directly. For the QR-LMS algorithm, $r_{i,i}(n-1)$'s are chosen to be 1, instead of recursively computed using the algorithm in [12], and

$$\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n) = w^2\mathbf{I} + \mathbf{x}_N(n)\mathbf{x}_N^T(n).$$

The condition numbers of $\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n)$ and the QR-LMS algorithm can be shown to be [14]

$$\begin{aligned} \text{cond}(\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n)) &= 1 + \left\| \frac{\mathbf{x}_N(n)}{w} \right\|^2 \\ \text{cond}(\boldsymbol{\Phi}(n)) &= \sqrt{1 + \left\| \frac{\mathbf{x}_N(n)}{w} \right\|^2} \end{aligned}$$

respectively. In the A-QR-LS algorithm, $r_{i,i}(n-1)$ is the square root of the estimated signal power associated with the i th element in the input signal vector. It helps to normalize the individual components in the signal vector so as to minimize the

eigenvalue spread. By the same token, we can replace $r_{i,i}(n-1)$ using other recursive signal power estimator such as

$$r_{i,i}(n) = \sqrt{\sigma_i^2(n)}$$

where

$$\sigma_i^2(n) = \lambda_i \sigma_i^2(n-1) + \{x_i(n)\}^2 \quad (31)$$

and λ_i is a positive forgetting factor between 0 and 1. It will be shown in the simulation section that this simple power estimate, when used to estimate the power of individual elements in the transformed signal vector, leads to faster convergence speed in the TA-QR-LS algorithm. Further, for adaptive filtering applications with single and wide-sense stationary input, the condition number in (30) can be evaluated analytically. First of all, notice that $r_{1,1}(n-1) = r_{2,2}(n-1) = \dots = r_{N,N}(n-1) = \sigma_x$, the square root of the signal power of the input. This allows us to simply (30) to

$$\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n) = (w\sigma_x)^2\mathbf{I} + \mathbf{x}_N(n)\mathbf{x}_N^T(n) \quad (32a)$$

which is a rank-one modification of a scaled identity matrix with characteristic polynomial

$$|\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n) - \lambda\mathbf{I}| = ((w\sigma_x)^2 - \lambda)^{N-1} \times ((w\sigma_x)^2 + \mathbf{x}_N^T(n)\mathbf{x}_N(n) - \lambda). \quad (32b)$$

The largest and smallest eigenvalues of $\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n)$ are thus $(w\sigma_x)^2 + \mathbf{x}_N^T(n)\mathbf{x}_N(n)$ and $(w\sigma_x)^2$, respectively. Accordingly, the condition numbers of $\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n)$ and the A-QR-LS algorithm are

$$\begin{aligned} \text{cond}(\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n)) &= \frac{((w\sigma_x)^2 + \mathbf{x}_N^T(n)\mathbf{x}_N(n))}{(w\sigma_x)^2} \\ &= 1 + \left\| \frac{\mathbf{x}_N(n)}{w\sigma_x} \right\|^2 \end{aligned} \quad (33a)$$

$$\begin{aligned} \text{cond}(\boldsymbol{\Phi}(n)) &= \sqrt{\text{cond}(\boldsymbol{\Phi}^T(n)\boldsymbol{\Phi}(n))} \\ &= \sqrt{1 + \left\| \frac{\mathbf{x}_N(n)}{w\sigma_x} \right\|^2} \end{aligned} \quad (33b)$$

respectively. Since σ_x is the square root of the average power of $\mathbf{x}_N(n)$, the average value of $\|\mathbf{x}_N(n)/(w\sigma_x)\|^2$ is thus equal to $1/w^2$. Therefore, the condition number of the A-QR-LS algorithm is less sensitive to the signal power of the input. For the TA-QR-LS algorithm, the power estimates $\sigma_i(n)$ will in general differ from each other. The exact condition number is quite difficult to analyze. However, the overall effect is still to reduce the sensitivity of the algorithm to variation of input signal power and eigenvalue spread.

IV. GIVENS-BASED APPROXIMATE QR-LS ALGORITHM

Although the Householder transformation is in general more efficient than Givens rotation when multiple input signal vectors are processed, it is less efficient than the Givens rotation when signal vectors are processed one at a time [15]. In order to combine the back solving and the matrix updating steps in the approximate QR-LS algorithm, the signal vector has to be processed one at a time. Hence, it is possible to reduce the arithmetic complexity further by employing the Givens rotation instead of the Householder transformation as proposed originally

in [12]. Furthermore, thanks to the Givens rotations, it is possible to develop a “square-root free” version of the resulting Givens-based $O(N)$ approximate QR-LS algorithm, similar to the classical work in [6]. This greatly reduces the complexity of the algorithm and is very easy to implement in hardware. For notation convenience, let us consider the triangularization of the following $(N+1) \times (N+1)$ matrix B (the shorthand of $\tilde{D}_N(n)$) having the same structure as in (20):

$$B = \begin{bmatrix} b_{1,1} & & & & b_{1,N+1} \\ & b_{2,2} & & \mathbf{0} & b_{2,N+1} \\ & \mathbf{0} & \ddots & & \vdots \\ & & & b_{N,N} & b_{N,N+1} \\ b_{N+1,1} & b_{N+1,2} & \cdots & b_{N+1,N} & b_{N+1,N+1} \end{bmatrix}. \quad (34)$$

Owing to the special structure of matrix B , only two rows, the i th row and the $(N+1)$ th row, are changed when applying Givens rotation to it during the i th iteration, i.e.,

$$b_{i,j}^{(k)} = \begin{cases} b_{i,j}, & \text{if } k \leq i \\ b_{i,j}^*, & \text{otherwise} \end{cases} \quad (35)$$

where $b_{i,j}^*$ denotes the element of the upper triangular matrix transformed. Following the derivation in [12], the following approximate QR-LS algorithm based on Givens rotations is obtained as shown in (36) at the bottom of the next page, where $\gamma_i = \sum_{j=i+1}^N b_{N+1,j} \theta_j(n)$. In order to avoid the square root operations in the Givens-based algorithm, a similar approach as in the square-root free Givens QRD algorithm [6] is derived below. More precisely, B is rewritten as follows in (37), shown at the bottom of the next page, where $\sqrt{\delta_i} = b_{i,i}$, $b_{i,N+1} = \sqrt{\delta_i} \tilde{b}_{i,N+1}$, $i = 1, 2, \dots, N$; $b_{N+1,j} = \alpha_0 \tilde{b}_{N+1,j}$, $j = 1, 2, \dots, N+1$. Consider the i th Givens rotation, where the $(N+1, j)$ -entries $\tilde{b}_{N+1,j}^{(i)}$ of the previously rotated matrix, $\mathbf{G}^{(1)} \mathbf{G}^{(2)} \dots \mathbf{G}^{(i-1)} \cdot B$ is being annihilated. (Please see (38) at the bottom of the next page.) For simplicity, only the i th and $(N+1)$ th rows of $\mathbf{G}^{(1)} \mathbf{G}^{(2)} \dots \mathbf{G}^{(i-1)} \cdot B$ are shown. From (38), we get

$$b_{N+1,j}^{(i)} = \sqrt{\alpha_{i-1}} \cdot \tilde{b}_{N+1,j}^{(i)}, \quad j = i+1, \dots, N+1 \quad (39)$$

where α_i , $i = 1, 2, \dots, N$ are auxiliary parameters given by $\alpha_0 = 1$, $\alpha_i = \alpha_{i-1} \delta_i / \delta_i^*$. Since the Givens rotation annihilates the first nonzero element of the second row, it can be shown that

$$\begin{aligned} \delta_i^* &= \delta_i + \alpha_{i-1} \left[\tilde{b}_{N+1,i}^{(i)} \right]^2, \quad c_i = \sqrt{\delta_i} / \sqrt{\delta_i^*} \\ s_i &= \sqrt{\alpha_{i-1}} \tilde{b}_{N+1,i}^{(i)} / \sqrt{\delta_i^*}. \end{aligned} \quad (40)$$

Combining (38) and (40), we have

$$\begin{aligned} \sqrt{\alpha_i} \tilde{b}_{N+1,j}^{(i+1)} &= \left(\sqrt{\delta_i} \sqrt{\alpha_{i-1}} \tilde{b}_{N+1,j}^{(i)} / \sqrt{\delta_i^*} \right) \\ &\quad + \left(\sqrt{\delta_i} \sqrt{\alpha_{i-1}} \tilde{b}_{i,j} \tilde{b}_{N+1,i}^{(i)} / \sqrt{\delta_i^*} \right) \end{aligned} \quad (41)$$

and $\sqrt{\delta_i^*} \tilde{b}_{i,j}^* = (\delta_i \tilde{b}_{i,j} / \sqrt{\delta_i^*}) + (\alpha_{i-1} \tilde{b}_{N+1,j} \tilde{b}_{N+1,i}^{(i)} / \sqrt{\delta_i^*})$. Because $\tilde{b}_{i,j}$ is equal to zero, (41) can be written after some algebra as

$$\tilde{b}_{N+1,j}^{(i+1)} = \tilde{b}_{N+1,j}^{(i)}, \quad \tilde{b}_{i,j}^* = \rho_i \tilde{b}_{N+1,j}^{(i)} \quad (42)$$

TABLE I
GIVENS-BASED TA-QR-LS ALGORITHM

1. Initialization		
$\theta = [0, 0, \dots, 0]^T$		
$\delta = [1, 1, \dots, 1]^T$, $\alpha_0 = 1$		$\mathbf{b}' = \{b_{N+1,i}\}$, $\mathbf{b} = \{b_{i,N+1}\}$
2. Recursive Operations		For complex inputs, modify
$\mathbf{b} = -\theta$, $\delta' = w\delta$		$\alpha' = \alpha_{i-1} b_{N+1,i}$ to $\alpha' = \alpha_{i-1} b_{N+1,i}^*$,
Add New Equation and DCT Transform		where * denotes complex conjugate.
$\mathbf{b}' = \text{DCT}(\mathbf{x}_d)$		
Transformation		
For $i = 1, 2, \dots, N$ Loop		
$\alpha' = \alpha_{i-1} b_{N+1,i}$; $\delta_i = \delta_i' + \alpha' b_{N+1,i}$		} $8N$ multiplications
$\sigma_i = \delta_i' / \delta_i$; $\rho_i = \alpha' / \delta_i$		
$\alpha_i = \alpha_{i-1} \sigma_i$; $\tau = b_{i,N+1}$		
$b_{i,N+1}^* = \sigma_i b_{i,N+1} + \rho_i b_{N+1,N+1}^{(i)}$		
$\tilde{b}_{N+1,N+1}^{(i+1)} = b_{N+1,N+1}^{(i)} - \tau b_{N+1,i}$		
End of Loop		} $3N$ additions
$\gamma_N = 0$, $\tilde{\theta}(N) = -b_{N,N+1}^*$		
For $i = N-1, N-2, \dots, 1$ Loop		
$\gamma_i = \gamma_{i+1} + b_{i,N+1} \theta(i+1)$		} $2(N-1)$ multiplications
$\tilde{\theta}(i) = -b_{i,N+1}^* - \rho_i \gamma_i$		
End of Loop i		} $2(N-1)$ additions
Invert DCT Transform		
$\theta = \text{IDCT}(\tilde{\theta})$		
3. Termination Control		
If $T.\text{Flag} = \text{True}$, Then Stop		
Else go to 2		

where $\rho_i = \alpha_{i-1} \tilde{b}_{N+1,i}^{(i)} / \delta_i^*$. Similarly, we have

$$\begin{aligned} \tilde{b}_{N+1,N+1}^{(i+1)} &= \tilde{b}_{N+1,N+1}^{(i)} - \tilde{b}_{i,N+1} \tilde{b}_{N+1,i}^{(i)} \\ \tilde{b}_{i,N+1}^* &= \sigma_i \tilde{b}_{i,N+1} + \rho_i \tilde{b}_{N+1,N+1}^{(i)} \end{aligned} \quad (43)$$

where $\sigma_i = \delta_i / \delta_i^*$. Hence, the square root operations are avoided in the updating equations in (42) to (43). Finally, we get the recursive parameter estimation formulas for the proposed square root-free Givens-based approximate QR-LS algorithm (TA-QR-LS) as shown in Table I. The Givens-based A-QR-LS algorithm only requires $10N$ multiplications and $5N$ additions. The arithmetic complexity of this algorithm is considerably lower than the original A-QR-LS algorithm in [12], which requires N square roots, $17N$ multiplications, and $9N$ additions. It should be noted that a similar approach can be applied to the QR-LMS algorithm in [13] (where all the $b_{i,i}$'s are equal to 1). Table II compares the arithmetic complexities of these algorithms. For simplicity, the divisions in the algorithms are counted as multiplications. $M.T(N)$ and $A.T(N)$ are respectively the multiplications and additions required for the transformation \mathbf{C} and inverse transformation \mathbf{C}^H . If \mathbf{C} is chosen as the DCT, fast algorithms are available for their computation [25]–[29]. If N is a power of two, fast algorithms [25], [26] with complexity

$$\begin{aligned} M.\text{DCT}(N) &= (N/2) \log_2 N \\ A.\text{DCT}(N) &= (3/2)N \log_2 N - N + 1 \end{aligned}$$

TABLE II
ARITHMETIC COMPLEXITIES OF THE VARIOUS ALGORITHMS

	Proposed Givens-based A-QR-LS	A-QR-LS in [13]	Proposed TA-QR-LS	QR-LMS in [14]	Proposed Givens-based QR-LMS
MULT	$10N$	$17N$	$2 \cdot M_T(N) + 10N$	$17N$	$10N$
ADD	$5N$	$9N$	$2 \cdot A_T(N) + 5N$	$9N$	$5N$
Square root	No	N	No	N	No

can be employed. Fast DCT algorithms for composite values of N are also available [28], [30], [32], and their complexities are similar to that of the power of two algorithm. For adaptive filtering applications with single input, the inverse DCT (IDCT) can be computed recursively in $4N$ additions and $2N$ multiplications [25]. Fast algorithms for computing the discrete

Fourier transforms (DFT) [30], [31] and discrete Hartley transforms (DHT) can be found in [28], [31]–[33]. Interested readers are referred to [29] for more details on other unitary transformations and their fast algorithms. Finally, we remark that the TA-QR-LS algorithm is very regular for systolic implementation. Due to page limitation, the details are omitted.

Upper Triangular Algorithm

$$\left\{ \begin{array}{l} \pi_0 = 1 \\ \text{For } i = 1, 2, \dots, N \text{ Loop} \\ \alpha_i = \sqrt{(b_{i,i})^2 + (\pi_{i-1} b_{N+1,i})^2}, \quad c = b_{i,i}/\alpha_i, \\ \rho = -\pi_{i-1} b_{N+1,i}/\alpha_i \\ \pi_i = \pi_{i-1} c, \quad \delta_i = -\rho \pi_{i-1}, \quad b_{i,i}^* = \alpha_i \\ b_{i,N+1}^* = c b_{i,N+1} - \rho b_{N+1,N+1}^{(i)}, \quad b_{N+1,N+1}^{(i+1)} = \rho b_{i,N+1} + c b_{N+1,N+1}^{(i)}, \\ \left\{ \begin{array}{l} \text{For } j = i + 1, i + 2, \dots, N \text{ Loop} \\ b_{i,j}^* = \delta_i b_{N+1,j} \\ \text{End of Loop } j \end{array} \right. \\ \text{End of Loop } i \end{array} \right. ,$$

Backsolving Algorithm

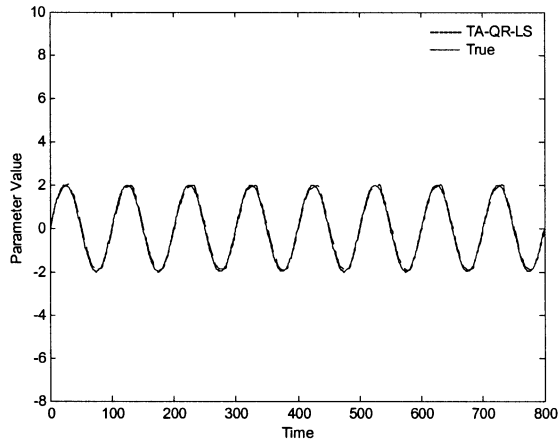
$$\begin{aligned} \gamma_N &= 0, \quad s_N = -b_{N,N+1}^*, \quad \theta_N(n) = s_N/b_{N,N}^* \\ \text{For } i &= N-1, N-2, \dots, 1 \text{ Loop} \\ \gamma_i &= \gamma_{i+1} + b_{N+1,i+1} \theta_{i+1}(n), \quad s_i = -b_{i,N+1}^* - \delta_i \gamma_i, \quad \theta_i(n) = s_i/b_{i,i}^* \\ \text{End of Loop } i \end{aligned} \quad (36)$$

$$B = \begin{bmatrix} \sqrt{\delta_1} & & & & \sqrt{\delta_1} \tilde{b}_{1,N+1} \\ & \sqrt{\delta_2} & & \mathbf{O} & \sqrt{\delta_2} \tilde{b}_{2,N+1} \\ \mathbf{O} & & \ddots & & \vdots \\ & & & \sqrt{\delta_N} & \sqrt{\delta_N} \tilde{b}_{N,N+1} \\ \alpha_0 \tilde{b}_{N+1,1} & \alpha_0 \tilde{b}_{N+1,2} & \cdots & \alpha_0 \tilde{b}_{N+1,N} & \alpha_0 \tilde{b}_{N+1,N+1} \end{bmatrix} \quad (37)$$

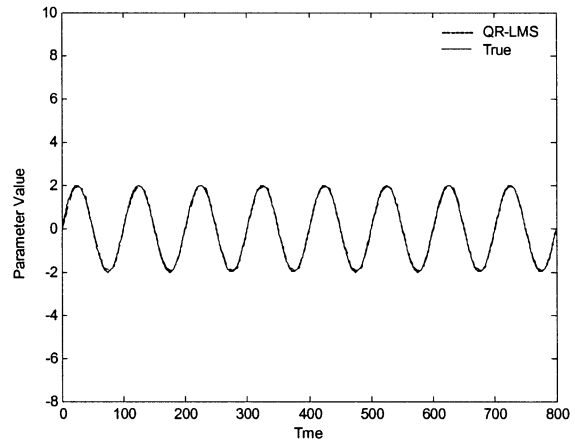
$$\begin{aligned} \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} \begin{bmatrix} 0 \cdots 0 & \sqrt{\delta_i} & 0 & \cdots & 0 & \sqrt{\delta_i} \tilde{b}_{i,N+1} \\ 0 \cdots 0 & \sqrt{\alpha_{i-1}} \tilde{b}_{N+1,i}^{(i)} & \sqrt{\alpha_{i-1}} \tilde{b}_{N+1,i+1}^{(i)} & \cdots & \sqrt{\alpha_{i-1}} \tilde{b}_{N+1,N}^{(i)} & \sqrt{\alpha_{i-1}} \tilde{b}_{N+1,N+1}^{(i)} \end{bmatrix} \\ = \begin{bmatrix} 0 \cdots 0 & \sqrt{\delta_i} & \sqrt{\delta_i} \tilde{b}_{i,i+1} & \cdots & \sqrt{\delta_i} \tilde{b}_{i,N} & \sqrt{\delta_i} \tilde{b}_{i,N+1}^* \\ 0 \cdots 0 & 0 & \sqrt{\alpha_i} \tilde{b}_{N+1,i+1}^{(i+1)} & \cdots & \sqrt{\alpha_i} \tilde{b}_{N+1,N}^{(i+1)} & \sqrt{\alpha_i} \tilde{b}_{N+1,N+1}^{(i+1)} \end{bmatrix}. \end{aligned} \quad (38)$$

TABLE III
MAXIMUM ERRORS OF THE ESTIMATES OBTAINED BY THE VARIOUS ALGORITHMS

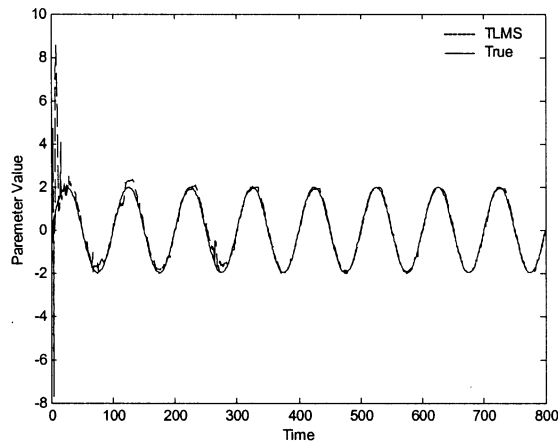
Parameters	$\sigma = 0.1$			$\sigma = 0.2$			$\sigma = 0.3$		
	TA-QR-LS	A-QR-LS	RLS	TA-QR-LS	A-QR-LS	RLS	TA-QR-LS	A-QR-LS	RLS
-1.96	0.00406	0.04999	0.00052	0.00691	0.05418	0.00150	0.00459	0.05154	0.00224
-0.98	0.00428	0.05316	0.00047	0.00621	0.05318	0.00129	0.00437	0.05094	0.00208
1.0	0.00670	0.01086	0.00070	0.02278	0.00205	0.00047	0.02265	0.00389	0.00114
2.0	0.02224	0.06597	0.00048	0.07924	0.05838	0.00238	0.09550	0.05309	0.00160
0.5	0.01139	0.03369	0.00065	0.04975	0.04651	0.00159	0.07336	0.05546	0.00267



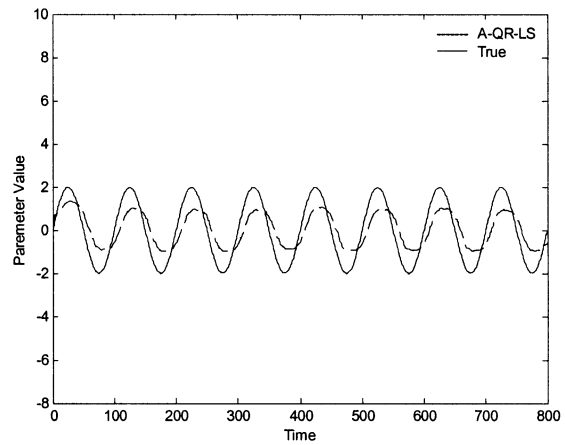
(a)



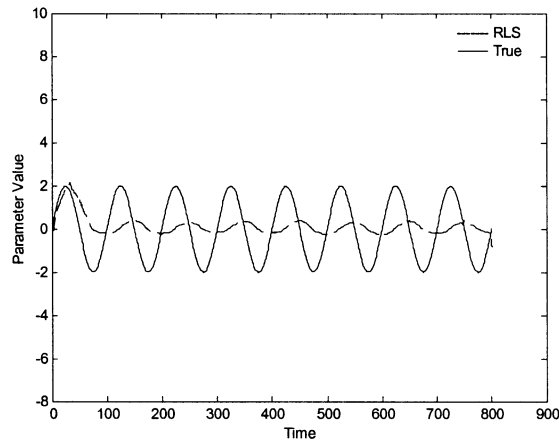
(b)



(c)



(d)



(e)

Fig. 1. True and estimated values of $a(k)$ obtained by: (a) TA-QR-LS; (b) QR-LMS; (c) TLMS; (d) A-QR-LS; and (e) RLS.

V. EXPERIMENTAL RESULTS

Example 1 System Identification: The first simulation is the identification of the following second-order system, which is used in [13]

$$y(k) = a(k)y(k-1) - 0.98y(k-2) + u(k) + 2.0u(k-1) + 0.5u(k-2) + \zeta(k) \quad (44)$$

where $u(k)$, $y(k)$, and $\zeta(k)$ are respectively the input, output, and measurement noise of the system. The system is driven by a white zero mean white Gaussian process $u(k)$ with unit variance. The measurement noise $\zeta(k)$ is assumed to be a zero mean white Gaussian noise process with variance σ^2 . In the first part of the simulation, $a(k)$ is chosen to be a constant equal to -1.96 . The objective is to estimate the model parameters from the system input and output for $k = 1, 2, \dots, 800$. Table III summarizes the maximum errors when estimating these parameters after the algorithms have converged. These results are obtained by averaging over 100 Monte Carlo simulations with a forgetting factor $w = 0.98$. Since the DCT is an efficient transformation for decorrelating signal and fast computational algorithms are available, it is employed in this study. The parameter vectors of all algorithms are initialized to zero. The inverse of the correlation matrix \mathbf{P} is initialized to $\delta^{-1}\mathbf{I}$, where $\delta = 0.01\sigma_{\text{input}}^2$, and σ_{input}^2 is the power of the input signal vector. From Table III, it can be seen that the steady state maximum errors of the estimates of the RLS algorithm is smaller than those of the TA-QR-LS and the A-QR-LS algorithms, which are comparable to each other. In the second part of the simulation, the parameter $a(k)$ is assumed to be time-varying, and is modeled by a slowly varying sinusoid: $a(k) = 2\sin(2\pi fk)$ with $f = 0.01$ Hz, to compare the tracking performance of the algorithms. The forgetting factors w for the TA-QR-LS, and A-QR-LS algorithms are both equal to 0.7. Because the correlation matrix of the input signal is nearly singular, the RLS algorithm cannot converge when the forgetting factor w is smaller than 0.97, a value of 0.98 is chosen. The weight factor w for the QR-LMS algorithm is chosen as 0.7 and the stepsize for the TLMS algorithm is chosen as 0.1. The estimates of $a(k)$ obtained by the TA-QR-LS, QR-LMS, TLMS, A-QR-LS and RLS algorithms are shown in Fig. 1. Very good performances are obtained by the TA-QR-LS and the QR-LMS algorithms, which are comparable to each other. There is a considerable parameter fluctuation in the TLMS algorithm at the beginning of the simulation, and its value start to settle down after simulation time $k = 300$. The A-QR-LS and the RLS algorithms are unable to track the slowly time-varying parameter, and the performance of the RLS algorithm seems to be the worse. As mentioned earlier, the main diagonal elements of $\mathbf{D}^2(n)$ in (22) of the TA-QR-LS algorithm are power estimates of the elements in the input signal vector. However, their values only converge slowly to the true estimates. Therefore, the tracking speed of the TA-QR-LS algorithm is almost identical to that of the QR-LMS algorithm. In order to evaluate the performance of the various algorithms to sudden change of system parameter, $a(k)$ is suddenly changed from -1.96 to -1.5 after $k > 780$. The results are shown in Fig. 2, and it was obtained using 100 Monte Carlo simulations. In order to improve the tracking speed of the TA-QR-LS algorithm, the initial values of the diagonal elements of $\mathbf{D}^2(n)$ for simulation time $k = 1$ to 8 are replaced by the power estimates as mentioned in (31). This improved algorithm is referred to as initial power

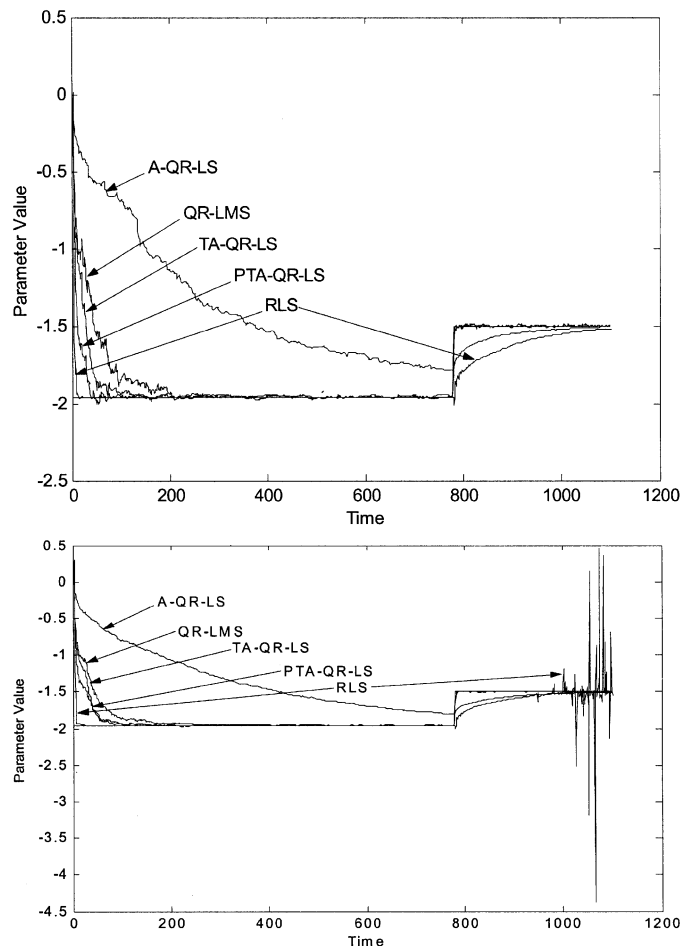


Fig. 2. True and estimated values of $a(k)$ by various algorithms.

estimated transform domain approximate QR-LS algorithm (PTA-QR-LS). The forgetting factors w for the TA-QR-LS, PTA-QR-LS, QR-LMS, and A-QR-LS algorithms are chosen as 0.32, to allow for a faster tracking speed. Because the correlation matrix of the input is nearly singular, the forgetting factor w for the RLS algorithm is chosen to be 0.98. It can be seen that the initial convergence of the RLS algorithm is better than the A-QR-LS, TA-QR-LS, PTA-QR-LS, and QR-LMS algorithms. However, its response to the sudden change of system parameter is rather poor because of the relatively large forgetting factor used to avoid the correlation matrix from being singular. Even so, the RLS algorithm exhibits instability after simulation time $k > 1000$. The performances of the QR-LMS, TA-QR-LS, and PTA-QR-LS algorithms are more satisfactory, with the best performance achieved by the PTA-QR-LS algorithm. It can be seen that the transformation and initial power estimation help to improve the convergence and tracking speed of the algorithms.

Example 2 Acoustic Echo Cancellation: The second simulation is to evaluate the performances of the various algorithms for acoustic echo cancellation. The input signal is an artificially generated speech signal, which is modeled as a fifth order AR model characterized by poles at: $z_1 = 0.5$, z_2 and $z_3 = 0.85e^{\pm j\pi/3}$, z_4 and $z_5 = 0.7e^{\pm j2\pi/3}$. 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

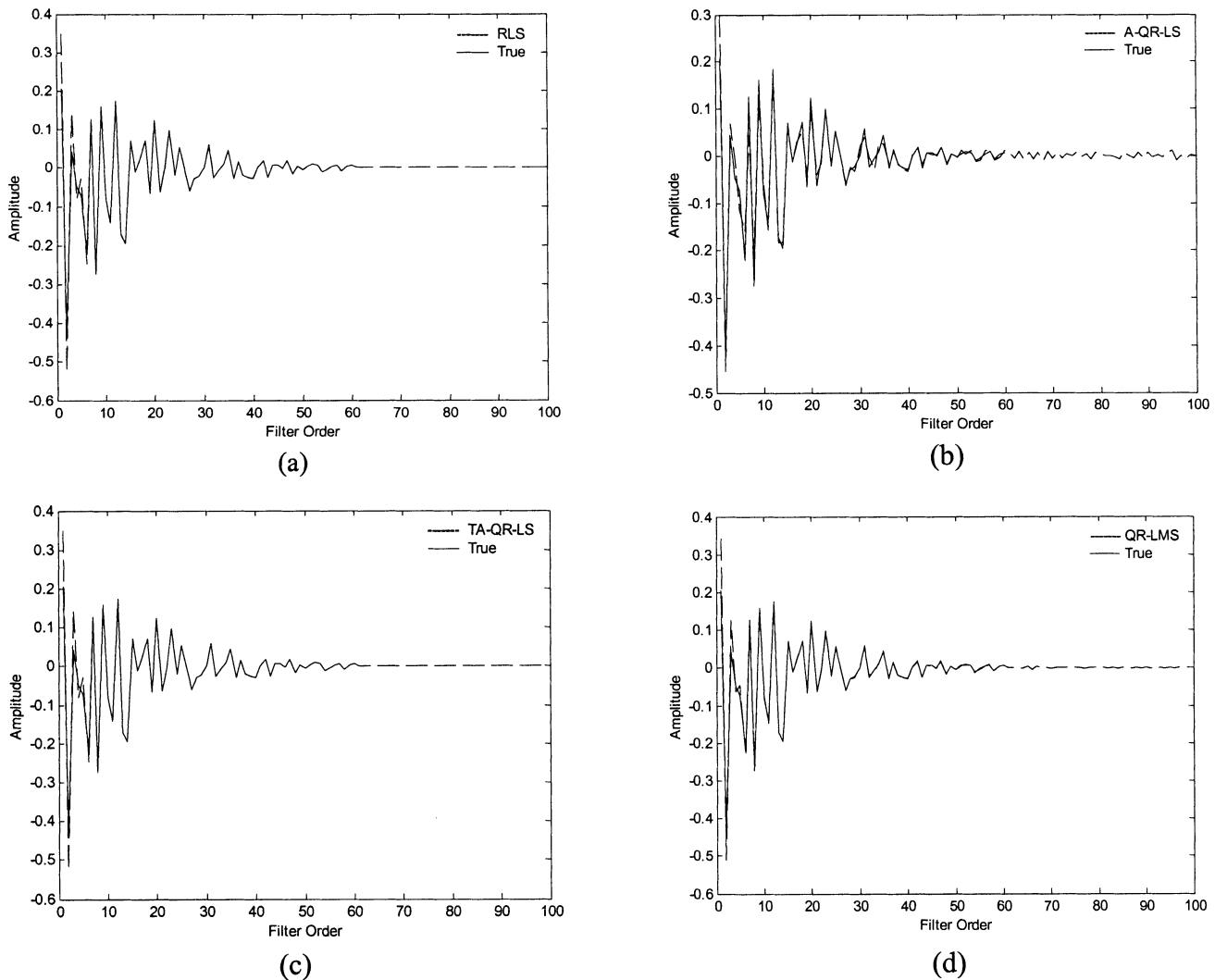


Fig. 3. Echo path estimated by various algorithms.

length of the acoustic path. Fig. 3 shows the estimated echo paths obtained by the TA-QR-LS, and A-QR-LS algorithms. The result for the PTA-QR-LS is very close to the TA-QR-LS in this case, and it is omitted for page limitation. The estimated echo path for the TA-QR-LS and RLS algorithms are similar to each other in appearance and the result for the RLS algorithm is omitted for simplicity. The error norms for all the algorithms are shown in Fig. 4. These estimates are obtained by averaging over 100 Monte Carlo simulations. The forgetting factors w for these three algorithms are chosen as 0.99. It can be seen that the trail coefficients after $n = 60$ of the echo path estimated by the A-QR-LS algorithm are considerable larger than the TA-QR-LS (and the RLS) algorithm, though the steady state error norms are almost identical for all the algorithms. From Fig. 4, it can be seen that the initial convergence of the RLS algorithm is inferior to the other algorithms. It is due to the large time lag required to forget the initial correlation matrix and to estimate the correct correlation matrix of large size (100×100). After the correlation matrix has been estimated, the RLS algorithm converges rapidly to the final solution. Another reason for the fast initial convergence of the LMS-based algorithms is that the echo path coefficients in the exponential decaying model are relatively small in amplitude. Therefore, the LMS algorithms are easier to converge to these

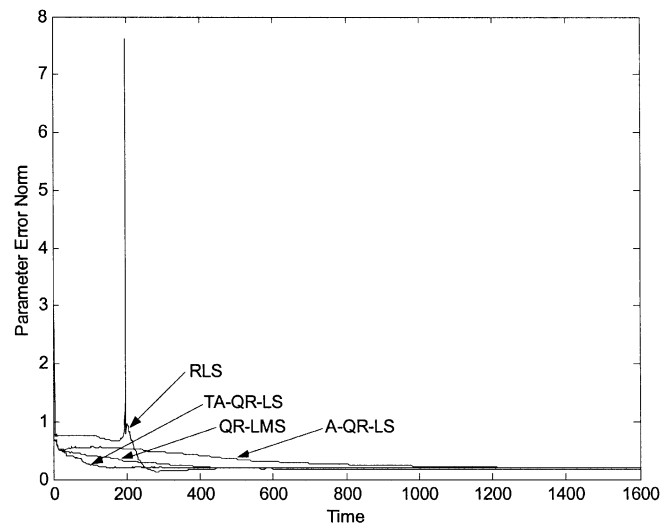


Fig. 4. Error norm of the parameter vector for various algorithms.

small coefficients with an initial zero weight vector. This implies that the proposed PTA-QR-LS algorithm is attractive in acoustic applications where large number of coefficients with small coefficients has to be estimated.

VI. CONCLUSION

Improved algorithms for a class of approximate QR-based LS (A-QR-LS) algorithms recently proposed by Liu [12] are presented. It is shown that the A-QR-LS algorithm is equivalent to an element-wise normalized LMS algorithm with time-varying stepsizes. It reduces to the QR-LMS algorithm when all the normalization constants are chosen as the Euclidean norm of the input signal vector. Two transform-domain approximate QR-LS algorithms, the TA-QR-LS, and the initial power estimated TA-QR-LS (PTA-QR-LS) algorithms, are proposed to improve the convergence speed by decorrelating the input signal vector with unitary transformations. The mean weight vectors of these algorithms are shown to converge to the optimal Wiener solution if the weighting factor w lies between 0 and 1. New realizations of the A-QR-LS, TA-QR-LS, PTA-QR-LS, and QR-LMS algorithms using the Givens rotations are presented. This reduces the arithmetic complexity by a factor of two and allows square root-free versions of the algorithms to be developed. Simulation results show that the proposed TA-QR-LS and the PTA-QR-LS algorithms are good alternatives to the RLS and QR-LMS algorithms in applications involving multiple channels, acoustic modeling, and fast parameter variations.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their comments, which greatly improve the presentation of this paper. They also thank and gratefully acknowledge the Hong Kong Research Grant Council.

REFERENCES

- [1] P. A. Regalia and M. G. Bellanger, "On the duality between fast QR methods and lattice methods in least squares adaptive filtering," *IEEE Trans. Signal Processing*, vol. 39, pp. 879–891, Apr. 1991.
- [2] M. T. Morf, T. Kailath, and L. Ljung, "Fast algorithms for recursive identification," in *Proc. 1976 Congress Decision Control*, Clearwater Beach, FL, 1976, pp. 916–921.
- [3] D. D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439–1446, 1978.
- [4] G. Carayannis, D. G. Manolakis, and N. Kalouptsidis, "A fast sequential algorithm for least squares filtering and prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1394–1402, 1983.
- [5] J. M. Cioffi and T. Kailath, "Fast RLS transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 304–337, 1984.
- [6] W. M. Gentleman and H. T. Kung, "Matrix triangularization by systolic array," in *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 298, 1981, pp. 298–303.
- [7] J. G. McWhirter, "Recursive least squares minimization using a systolic array," in *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 431, 1983, pp. 105–112.
- [8] J. M. Cioffi, "A fast adaptive ROTOR's RLS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 631–653, 1990.
- [9] S. Ljung and L. Ljung, "Error propagation properties of recursive least squares adaptation algorithms," *Automatica*, vol. 21, pp. 157–167, 1985.
- [10] J. M. Cioffi, "Limited precision effects in adaptive filtering," *IEEE Trans. Circuits Syst.*, vol. 34, pp. 821–833, July 1987.
- [11] D. T. M. Slock and T. Kailath, "Numerically stable fast recursive least squares transversal filters," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 1369–1372, 1988.
- [12] Z. S. Liu, "QR methods of $O(N)$ complexity in adaptive parameter estimation," *IEEE Trans. Signal Processing*, vol. 43, pp. 720–729, 1995.
- [13] 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, Mar. 1998.
- [14] S. H. Leung and C. F. So, "Nonlinear RLS algorithm using variable forgetting factor in mixture noise," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3777–3780, 2001.
- [15] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias, *Advanced Digital Signal Processing*. New York: Macmillan, 1992.
- [16] F. B. Boroujeny, *Adaptive Filters: Theory and Applications*. New York: Wiley, 1999.
- [17] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2001.
- [18] G. Glentis, K. Berberidis, and S. Theodoridis, "Efficient least squares adaptive algorithms for FIR transversal filtering," *IEEE Signal Processing Mag.*, pp. 13–41, July 1999.
- [19] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [20] M. Hartenneck, J. G. McWhirter, I. K. Proudler, and R. W. Stewart, "Algorithmically engineered fast multichannel adaptive filter based on QR-RLS," in *Proc. Inst. Elect. Eng. Visual Image Signal Processing*, vol. 146, Feb. 1999.
- [21] G. Glentis and N. Kalouptsidis, "Fast adaptive algorithms for multichannel filtering and system identification," *IEEE Trans. Signal Processing*, vol. 40, pp. 2433–2458, Oct. 1992.
- [22] —, "Efficient order recursive algorithms for multichannel least squares filtering," *IEEE Trans. Signal Processing*, vol. 40, pp. 1354–1374, June 1992.
- [23] J. G. McWhirter and I. K. Proudler, "The QR family," in *Adaptive System Identification and Signal Processing*, N. Kalouptsidis and S. Theodoridis, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1993, ch. 7, pp. 260–321.
- [24] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, "Computationally efficient QR decomposition approach to least squares adaptive filtering," *Proc. Inst. Elect. Eng.—F*, vol. 138, pp. 341–353, Aug. 1991.
- [25] S. C. Chan and K. L. Ho, "Direct methods for computing the discrete sinusoidal transforms," *Proc. Inst. Elect. Eng.*, pt. F, vol. 137, pp. 433–442, Dec. 1990.
- [26] —, "A new 2-D fast cosine transform algorithm," *IEEE Trans. Signal Processing*, vol. 29, pp. 481–485, Feb. 1991.
- [27] —, "Fast algorithms for computing the discrete cosine transforms," *IEEE Trans. Circuits Syst. II*, vol. 49, pp. 185–190, Mar. 1992.
- [28] —, "Split vector-radix fast Fourier transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2029–2039, Aug. 1992.
- [29] P. R. Yip and K. R. Rao, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York: Academic, 1990.
- [30] C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms*. New York: Wiley, 1985.
- [31] S. C. Chan and K. L. Ho, "Prime factor real-valued Fourier, cosine and Hartley transforms," in *Proc. EUSIPCO 92*, Brussels, Belgium, Aug. 24–27, 1992, pp. 1045–1048.
- [32] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, pp. 803–816, Aug. 1994.
- [33] S. C. Chan and K. L. Ho, "Fast algorithms for computing the discrete W transforms," in *Proc. IEEE Region 10 Conf. Computer and Communication Systems (TENCON)*, Hong Kong, Sept. 24–27, 1990, pp. 183–186.
- [34] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, "Transform domain LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 609–615, June 1983.
- [35] S. C. Douglas and H. Y. Meng, "Normalized data nonlinearities for LMS adaptation," *IEEE Trans. Signal Processing*, vol. 42, pp. 1352–1365, June 1994.
- [36] G. H. Golub and C. F. Van Loan, *Matrix Computations*, ser. Johns Hopkins Mathematical Sciences. Baltimore, MD: The John Hopkins Univ. Press, 1996.
- [37] F. Ling, "Efficient and robust least-squares lattice algorithms based on Givens rotation with systolic array implementation," in *Proc. 1989 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, San Diego, CA, Aug. 1989.
- [38] —, "Givens rotation based least-squares lattice and related algorithms," *IEEE Trans. Signal Processing*, vol. 39, pp. 1541–1551, July 1991.
- [39] F. Ling, D. Manolakis, and J. G. Proakis, "Numerically robust least-squares lattice-ladder algorithms with direct updating of the reflection coefficients," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 837–845, Aug. 1986.
- [40] M. D. Miranda and M. Gerken, "A hybrid least squares QR-lattice algorithm using a priori errors," *IEEE Trans. Signal Processing*, vol. 45, pp. 2900–2911, Dec. 1997.
- [41] A. A. Rontogiannis and S. Theodoridis, "New fast QR decomposition least squares adaptive algorithms," *IEEE Trans. Signal Processing*, vol. 46, pp. 2113–2122, Aug. 1998.
- [42] M. Bhouiri, 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.