FAST RECURSIVE EIGENSUBSPACE ADAPTIVE FILTERS

Peter Strobach

Fachhochschule Furtwangen, Box 763, Gerwigstr. 11
78120 Furtwangen, Germany

Abstract — A class of adaptive filters based on sequential eigendecomposition of the data covariance matrix is introduced. These new algorithms are completely rank revealing and hence they can perfectly handle the following two relevant data cases where conventional RLS methods fail to provide satisfactory results: 1) Highly oversampled "smooth" data with rank deficient or almost rank deficient covariance matrix. 2) Noise-corrupted data where a signal must be separated effectively from superimposed noise. This paper corrects the widely held believe that eigenbased algorithms must be computationally more demanding than conventional RLS techniques. A spatial RLS adaptive filter has a principal complexity of $O(N^2)$ operations per time step, where N is the filter order. Somewhat ironically, though, the corresponding new eigensubspace or low rank adaptive filter requires only O(Nr) operations per time step where $r \le N$ denotes the numerical rank of the data covariance matrix. Thus eigensubspace adaptive filters can be computationally less or even much less demanding depending on the rank/order ratio r/N or the "compressability" of the signal. Some high-performance subspace trackers are obtained as by-products of this research. Simulation results confirm our claims.

1. INTRODUCTION

Classical RLS techniques are based on two fatal assumptions: 1) They assume that the data covariance matrix has full rank $r = N$ and 2) they assume that the data is all signal and contains no noise. Such ideal conditions are seldom observed in practice. In communications and in sensor array processing, the signal is often buried in noise of considerable variance. Additionally, the numerical rank r of the signal covariance matrix is often smaller or much smaller than the number of observations or the filter order N. In these cases, the relevant information in a signal can be mapped or "compressed" with little or no loss of information into a dominant signal subspace of dimension $r \le N$ spanned by the orthogonal "eigensignals" or "natural modes" of the signal. In this paper, we develop the theory for a class of adaptive filters that project signals onto a dominant signal subspace rather than on the "complete" data subspace. These eigensubspace adaptive filters can handle both oversampled and noise-corrupted data. An effective separation of signal and noise is achieved and even a significant complexity reduction is possible when the number of sensors or taps is much larger than the dominant signal subspace dimension r. Thus the necessary amount of computations which must be expended in a specific application is no longer fixed, but is a function of the compressability of the given signal determined by the rank/order ratio r/N. Thus eigensubspace adaptive filtering means noise suppression *and* complexity reduction. The theory of eigensubspace or low rank adaptive filters is based on a mathematical concept named the Schur pseudo-inverse. This pseudo-inverse has the distinct property that it approximates the more well-known Moore-Penrose pseudo-inverse but can be updated recursively in time using reliable, stable and economical schemes. Results appear considerably condensed in this paper due to space limitations. The interested reader is referred to the detailed discussion given in [1].

2. RANK REDUCTION FOR ADAPTIVE FILTERING

Conventional RLS adaptive filters can be named "full rank" techniques because they project, at each time step, a reference signal vector $\mathbf{y}(t)$ onto a subspace spanned by the N column vectors of an $L \times N$ data matrix $\mathbf{X}(t) = [\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_N(t)]$. Define an orthogonal projection operator $\mathbf{P}_N(t)$,

$$\mathbf{P}_N(t) = \mathbf{X}(t)\mathbf{\Phi}^{-1}(t)\mathbf{X}^T(t) \quad , \tag{1}$$

and its orthogonal complement $\mathbf{P}_N^{\perp}(t) = \mathbf{I} - \mathbf{P}_N(t)$ where $\mathbf{I}$ is an $L \times L$ identity matrix and $\mathbf{\Phi}(t) = \mathbf{X}^T(t)\mathbf{X}(t)$ is an $N \times N$ sample covariance matrix. Assume first that the inverse of $\mathbf{\Phi}(t)$ exists. Then $\mathbf{d}(t)$ is the orthogonal projection of $\mathbf{y}(t)$ onto the column space of $\mathbf{X}(t)$ and $\mathbf{e}(t)$ is the orthogonal complement or "residual vector":

$$\mathbf{d}(t) = \mathbf{P}_N(t)\mathbf{y}(t) \quad , \qquad \mathbf{e}(t) = \mathbf{P}_N^{\perp}(t)\mathbf{y}(t) \quad . \tag{2a,b}$$

At each time step, an adaptive filter extracts only a single component of either $\mathbf{d}(t)$ or $\mathbf{e}(t)$. Usually the top components denoted by d(t) and e(t) are computed. They can be extracted from the vectors using a top pinning vector $\boldsymbol{\pi}^T(t) = [1, 0 \ldots 0]$:

$$d(t) = \boldsymbol{\pi}^T(t)\mathbf{d}(t) \quad , \qquad e(t) = \boldsymbol{\pi}^T(t)\mathbf{e}(t) \quad . \tag{3a,b}$$

A pinning of the data matrix extracts the actual data snapshot vector $\mathbf{z}(t)$,

$$\mathbf{z}^T(t) = \left[x_1(t), x_2(t), \ldots, x_N(t)\right] = \boldsymbol{\pi}^T(t)\mathbf{X}(t) \quad , \tag{4}$$

and a pinning of $\mathbf{y}(t)$ extracts the actual reference signal sample y(t). Introduce a cross correlation vector $\mathbf{c}(t)$ and the adaptive filter weight vector $\mathbf{a}(t)$:

$$\mathbf{c}(t) = \mathbf{X}^T(t)\mathbf{y}(t) \quad , \qquad \mathbf{a}(t) = \mathbf{\Phi}^{-1}(t)\mathbf{c}(t) \quad . \tag{5a,b}$$

Realize that:

$$d(t) = \mathbf{z}^T(t)\mathbf{a}(t) \quad , \qquad e(t) = y(t) - \mathbf{z}^T(t)\mathbf{a}(t) \quad , \tag{6a,b}$$

because $\mathbf{y}(t) = \mathbf{d}(t) + \mathbf{e}(t)$. Thus the top components d(t) and e(t) can be computed at each time step using a transversal filter or adaptive linear combiner of length N taps with coefficient vector $\mathbf{a}(t)$ and state vector $\mathbf{z}(t)$. In what follows, spatial and temporal adaptive filtering will be covered by the same unified low rank adaptive filter theory yielding fast algorithms for both cases.

Begin with the exact EVD of $\mathbf{\Phi}(t)$ as follows:

$$\mathbf{\Phi}(t) = \mathbf{V}(t)\mathbf{\Lambda}(t)\mathbf{V}^T(t) \quad , \tag{7}$$

where $\mathbf{V}(t) = [\mathbf{v}_1(t), \mathbf{v}_2(t), \ldots, \mathbf{v}_N(t)]$ is the $N \times N$ orthonormal matrix of eigenvectors and $\mathbf{\Lambda}(t) = \mathrm{diag}\left(\lambda_1(t), \lambda_2(t), \ldots, \lambda_N(t)\right)$ is the $N \times N$ diagonal matrix of eigenvalues. Assume that the eigenvalues in $\mathbf{\Lambda}(t)$ appear in the following magnitude structure:

$$\lambda_1(t) \ge \lambda_2(t) \ge \ldots \ge \lambda_r(t) \ge \lambda_{r+1}(t) = \sigma^2(t) = \ldots = \sigma^2(t) = \lambda_N(t) \,. \tag{8}$$

Here $\sigma^2(t)$ represents the noise floor or "water pouring" level. We distinguish two relevant data cases:

A: $\sigma^2(t)$ = small: "clean" and overmodeled signal.

B: $\sigma^2(t)$ = large : overmodeled signal in white noise.

In any case, we assume that the signal of interest can be concentrated in a dominant signal subspace of dimension $r \le N$ represented by the first r eigenvectors $\mathbf{V}_r(t) = [\mathbf{v}_1(t), \mathbf{v}_2(t), \ldots, \mathbf{v}_r(t)]$. Assume further that signal and noise are mutually orthogonal functions. Then the estimated data covariance matrix $\mathbf{\Phi}(t)$ can be represented as the sum of a rank r approximant $\mathbf{\Phi}_r(t) = \mathbf{V}_r(t)\mathbf{\Lambda}_r(t)\mathbf{V}_r^T(t)$ which carries the information about the signal of interest and a noise covariance matrix $\mathbf{\Phi}_{noise}(t)$:

$$\mathbf{\Phi}(t) = \mathbf{\Phi}_r(t) + \mathbf{\Phi}_{noise}(t) \tag{9}$$

Define the Moore-Penrose pseudo-inverse $\mathbf{\Phi}_r^{+}(t)$ of $\mathbf{\Phi}_r(t)$ as follows [2]:

$$\Phi_r^+(t) = V_r(t)\Lambda_r^{-1}(t)V_r^T(t) \quad , \tag{10}$$

where $\Lambda_r^{-1}(t) = \mathrm{diag}\left(\lambda_1^{-1}(t), \lambda_2^{-1}(t), \ldots, \lambda_r^{-1}(t)\right)$. A low rank or eigensubspace coefficient vector $a^+(t)$ is next defined by replacing the inverse covariance matrix in (5b) by the Moore-Penrose pseudo-inverse yielding the following low rank in-space and residual vectors $d^+(t)$ and $e^+(t)$, respectively:

$$d^+(t) = X(t)a^+(t) = X(t)V_r(t)\Lambda_r^{-1}(t)V_r^T(t)c(t) \quad , \tag{11a}$$

$$e^+(t) = y(t) - d^+(t) \quad . \tag{11b}$$

Recall that in (2a) the conventional (full rank) in-space vector $d(t)$ has been posed as the orthogonal projection of $y(t)$ onto the full column space of $X(t)$. Now, we can show (see [1] for details) that $d^+(t)$ can be interpreted as the orthogonal projection of $y(t)$ onto a dominant signal subspace determined by the first r left singular vectors in $U(t)$, where $X(t) = U(t)S(t)V^T(t)$ ist the SVD of $X(t)$. Thus define a projection operator $P_r(t) = U_r(t)U_r^T(t)$ that projects vectors orthogonally onto the dominant signal subspace spanned by the first r column vectors $U_r(t)$ in $U(t)$ to find:

$$d^+(t) = P_r(t)y(t) \quad , \qquad e^+(t) = P_r^{\perp}(t)y(t) \quad , \tag{12a,b}$$

where $P_r^{\perp}(t) = I - P_r(t)$. These expressions specify the eigensubspace projections corresponding to the conventional "full-subspace" projections according to (2a,b).

3. THE SCHUR PSEUDO-INVERSE

The computation of eigensubspace projections such as $d^+(t)$ and $e^+(t)$ requires knowledge about the dominant r eigenvectors $V_r(t)$ and the corresponding dominant r eigenvalues $\Lambda_r(t)$ of the data covariance matrix $\Phi(t)$ at each time step. One could update these components using recursive schemes [3]. The direct updating of an EVD, however, is a difficult task because the EVD is a strictly diagonal decomposition. Therefore, more profitable approaches start from a decomposition where the matrix of eigenvalues is not a priori restricted to diagonal shape, but can "blow up" to triangular structure in case of misadjustment. If defined appropriately, decompositions of this kind can be updated easily in time and their elements will converge rapidly towards the elements of the true EVD. For this purpose, consider the following "Schur-type" decomposition:

$$R(t) = Q^T(t)\Phi(t)Q(t-1) \tag{13}$$

Here $Q(t)$ is a strictly orthonormal N×r recursion matrix. $R(t)$ is an upper-right triangular matrix. Note that (13) originates from a pre-multiplication of both sides of $Q(t)R(t) = \Phi(t)Q(t-1)$ by $Q^T(t)$ because $Q^T(t)Q(t) = I$. Thus we may compute subsequent orthonormal recursion subspaces Q and corresponding triangular matrices R by the following two-term recurrence known as simultaneous orthogonal iteration [2]:

$$A(t) = \Phi(t)Q(t-1) \quad , \tag{14a}$$

$$A(t) = Q(t)R(t) \quad : \quad \text{"skinny" QR factorization} \tag{14b}$$

An auxiliary matrix $A(t)$ is formed as the product of $\Phi(t)$ and $Q(t-1)$. The desired $Q(t)$ and $R(t)$ are determined by the skinny QR factorization of $A(t)$. Simultaneous orthogonal iteration originates as a special case of Bauer's classical bi-iteration [4]. Provided that $\Phi(t)$ is time invariant with monotonically decreasing dominant eigenvalues one can show that the sequence of Q's will converge against the dominant eigensubspace $V_r(t)$ and the sequence of R's will converge against the diagonal matrix $\Lambda_r(t)$ of dominant eigenvalues. In our case, both $\Phi(t)$ and $c(t)$ are slowly varying functions of time because they are updated continuously according to:

$$\Phi(t) = \alpha\Phi(t-1) + (1-\alpha)z(t)z^T(t) \quad , \tag{15a}$$

$$c(t) = \alpha c(t-1) + (1-\alpha)z(t)y(t) \quad , \tag{15b}$$

where α is a positive exponential forgetting factor close to 1. In this case, the orthogonal iteration (14a,b) is a device that tracks the dominant eigensubspace $V_r(t)$ and the associated dominant eigenvalues in $\Lambda_r(t)$. Thus at each time step, the Schur-type decomposition (13) will tend to approximate $\Lambda_r(t) = V_r^T(t)\Phi(t)V_r(t)$. Consequently a matrix $\hat{\Phi}_r(t)$ defined as

$$\hat{\Phi}_r(t) = Q(t)R(t)Q^T(t-1) \tag{16}$$

will tend to approximate the low rank covariance matrix $\Phi_r(t) = V_r(t)\Lambda_r(t)V_r^T(t)$. We further conclude that the Moore-Penrose pseudo-inverse $\Phi_r^+(t)$ defined in (10) can be approximated by the Schur pseudo-inverse $\hat{\Phi}_r^+(t)$ defined as follows:

$$\hat{\Phi}_r^+(t) = Q(t-1)R^{-1}(t)Q^T(t) \quad . \tag{17}$$

Note that $R^{-1}(t)$ always exists provided that r is the rank of $\Phi(t)$. Introduce a projection operator $P_Q(t) = Q(t)Q^T(t)$ that projects vectors orthogonally onto the recursion subspace $Q(t)$ to demonstrate that the Schur pseudo-inverse satisfies the following relations: $\hat{\Phi}_r(t)\hat{\Phi}_r^+(t) = P_Q(t)$, $\hat{\Phi}_r^+(t)\hat{\Phi}_r(t) = P_Q(t-1)$. Use these relations to prove that the Schur pseudo-inverse satisfies the following set of Moore-Penrose conditions:

$$\hat{\Phi}_r(t)\hat{\Phi}_r^+(t)\hat{\Phi}_r(t) = \hat{\Phi}_r(t) \ , \qquad \hat{\Phi}_r^+(t)\hat{\Phi}_r(t)\hat{\Phi}_r^+(t) = \hat{\Phi}_r^+(t) \ ,$$

$$\left(\hat{\Phi}_r(t)\hat{\Phi}_r^+(t)\right)^T = \hat{\Phi}_r(t)\hat{\Phi}_r^+(t) \ , \quad \left(\hat{\Phi}_r^+(t)\hat{\Phi}_r(t)\right)^T = \hat{\Phi}_r^+(t)\hat{\Phi}_r(t) \ .$$

4. ADAPTIVE FILTERING USING THE SCHUR PSEUDO-INVERSE

Replace the Moore-Penrose pseudo-inverse in (11a,b) by the Schur pseudo-inverse (17) to obtain the following eigensubspece adaptive filter based on the Schur pseudo-inverse:

$$\hat{d}^+(t) = z^T(t)\hat{a}^+(t) = z^T(t)Q(t-1)R^{-1}(t)Q^T(t)c(t) \quad , \tag{18a}$$

$$\hat{e}^+(t) = y(t) - \hat{d}^+(t) \quad . \tag{18b}$$

The recursion matrix Q acts like a data compressor on both $z(t)$ and $c(t)$. Thus define "compressed" r×1 vectors $h(t)$, $g(t)$ and $g^*(t)$ as follows:

$$h(t) = Q^T(t-1)z(t) \, , \; g(t) = Q^T(t-1)c(t) \, , \; g^*(t) = Q^T(t)c(t) \, . \tag{19a-c}$$

Observe that the adaptive filter for the in-space component (18a) can be expressed in terms of $h(t)$ and $g^*(t)$:

$$\hat{d}^+(t) = h^T(t)R^{-1}(t)g^*(t) \quad . \tag{20}$$

A quick inspection of (20) reveals that this expression can be reduced to the computation of an inner product $\hat{d}^+(t) = h^T(t)p(t)$ with an auxiliary r×1 vector $p(t)$ which is obtained via back-substitution from the r×r upper triangular system $R(t)p(t) = g^*(t)$.

5. FAST EIGENSUBSPACE ADAPTIVE FILTERS USING A DECOMPOSITION OF PROJECTIONS

Fast subspace adaptive filtering requires a fast scheme for sequential orthogonal iteration. The key step towards a fast algorithm for sequential orthogonal iteration is the orthogonal projection of the actual recursion subspace $Q(t)$ onto the previous (one time step delayed) subspace $Q(t-1)$. Hereby $Q(t)$ is decomposed into an "in-space" component representing the "old" information in $Q(t)$ and an orthogonal complement subspace $\Delta(t)$ of dimension N×r that represents the innovation in $Q(t)$ based on the actual observation $z(t)$. Recall that the projection operator $P_Q(t-1)$ can be used to project vectors orthogonally onto $Q(t-1)$. Thus an orthogonal decomposition of $Q(t)$ can be stated as follows:

$$Q(t) = P_Q(t-1)Q(t) + \Delta(t) \quad , \tag{21}$$

where $P_Q(t-1)Q(t)$ is the information in $Q(t)$ which can be represented in the "old" subspace $Q(t-1)$ and $\Delta(t)$ is an innovations subspace that is orthogonal with respect to $Q(t-1)$. Thus $\Delta(t)$ satisfies: $Q^T(t-1)\Delta(t) = 0$. Introduce an r×r matrix $\Theta(t)$ of cosines of angles between subsequent subspaces as follows: $\Theta(t) =$

$Q^T(t-1)\,Q(t)$. Verify that the in-space component of $Q(t)$ can be expressed as a "rotated" version of $Q(t-1)$ because $P_Q(t-1)Q(t) = Q(t-1)Q^T(t-1)\,Q(t) = Q(t-1)\,\Theta(t)$. Thus the "cosine matrix" $\Theta(t)$ acts as a weight factor or rotor on the subspace $Q(t-1)$. A combination of the foregoing results finally yields:

$$Q(t) = Q(t-1)\,\Theta(t) + \Delta(t) \quad . \tag{22}$$

Continue with a substitution of the covariance time update (15a) into the "mapping equation" (14a) of orthogonal iteration:

$$\begin{aligned} A(t) &= \left[\alpha\Phi(t-1) + (1-\alpha)z(t)z^T(t)\right]Q(t-1) \\ &= \alpha\Phi(t-1)Q(t-1) + (1-\alpha)z(t)h^T(t) \quad . \end{aligned} \tag{23}$$

Express $Q(t-1)$ in (23) in terms of $Q(t-2)\Theta(t-1)$ and $\Delta(t-1)$ according to (22) to obtain: $A(t) = \alpha\Phi(t-1)Q(t-2)\Theta(t-1) + \alpha\Phi(t-1)\,\Delta(t-1) + (1-\alpha)z(t)h^T(t)$. Note that $\Phi(t-1)Q(t-2) = A(t-1)$ according to (14a) and therefore: $A(t) = \alpha A(t-1)\,\Theta(t-1) + \alpha\Phi(t-1)\,\Delta(t-1) + (1-\alpha)z(t)h^T(t)$. A surprising fact with this expression is that the term $\Phi(t-1)\,\Delta(t-1)$ must vanish completely provided only that signal and noise are mutually orthogonal and the dominant signal subspace dimension r has been chosen sufficiently large to accomodate all fundamental components or "natural modes" of the signal in $\hat{\Phi}_r$. The proof of this statement is long and appears in [1]. In any case, this result is fundamental because it states that auxiliary matrices A can be updated directly in time without any simplification via the following matrix recursion:

$$A(t) = \alpha A(t-1)\,\Theta(t-1) + (1-\alpha)z(t)h^T(t) \quad . \tag{24}$$

We are now in a position to establish a first variant of an eigensubspace or low rank adaptive filter. This algorithm named LORAF 1 consists of the following sequence of computations:

Initialize: $Q^T(0) = \left[\,I\ ,\ 0\,\right]$; $\Theta(0) = I$; $c^T(0) = [0\ldots0]$

For each time step compute: (19a), (24), (15b), (14b),
$\Theta(t) = Q^T(t-1)\,Q(t)$, (19c), $p(t) = R^{-1}(t)g^*(t)$,
$\hat{d}^+(t) = h^T(t)\,p(t)$, (18b).

LORAF 1 requires only $O(Nr^2)$ computations. The operations count is dominated by the explicit QR factorization in (14b). The question is: Is it really necessary to work with an auxiliary matrix $A(t)$ and its explicit QR factorization ? In fact the answer is NO ! One of the probably most striking results in this context is that Q and R factors can be updated *separately* in time. The auxiliary matrix $A(t)$ must not be formed explicitly anymore. This insight forms the basis for a class of sophisticated fast sequential eigensubspace adaptive filters and subspace trackers.

6. FAST EIGENSUBSPACE ADAPTIVE FILTERS BASED ON SEPARATE QR FACTOR TRACKING

The problem of interest is the direct time recursive updating of the QR factors of $A(t)$ for a sequence of A's generated according to (24). A somewhat simplified problem is the tracking of the QR factors of a sequence of A's generated according to

$$A(t) = \alpha A(t-1) + (1-\alpha)z(t)h^T(t) \quad . \tag{25}$$

Let's solve this problem first. Subsequently, we generalize the result and establish the exact QR factor tracker for the true sequence of A's generated by recursion (24). Consider (25) for the moment and note that the QR factors of $A(t)$ can be posed directly as a function of the QR factors of $A(t-1)$ plus a rank-one update as follows: $Q(t)R(t) = \alpha\,Q(t-1)R(t-1) + (1-\alpha)z(t)h^T(t)$.

We shall show that this updating problem has a fast solution. Only $3r-3$ Givens row rotations are finally required to compute the factors $Q(t)$ and $R(t)$ from their predecessors $Q(t-1)$ and $R(t-1)$. The auxiliary matrix $A(t)$ will not be formed explicitly anymore. A drastic reduction of the overall operations count will be achieved. Recall that the projection operator $P_Q(t)$ projects vectors orthogonally onto the subspace spanned by $Q(t)$. Hence the complement $z_\perp(t)$ of the orthogonal projection of $z(t)$ onto $Q(t-1)$ is computed as follows:

$$z_\perp(t) = P_Q^\perp(t-1)z(t) = \left(I - P_Q(t-1)\right)z(t) = z(t) - Q(t-1)h(t) \, . \tag{26}$$

Introduce the "energy" $Z(t)$ of $z_\perp(t)$: $Z(t) = z_\perp^T(t)\,z_\perp(t)$. Establish the normalized vector $\bar{z}_\perp(t)$ as follows:

$$\bar{z}_\perp(t) = Z^{-1/2}(t)\,z_\perp(t) = Z^{-1/2}(t)\,z(t) - Z^{-1/2}(t)\,Q(t-1)\,h(t) \, . \tag{27}$$

Next rearrange (27) to see that $z(t)$ can be expressed as the sum of $Z^{1/2}(t)\,\bar{z}_\perp(t)$ and $Q(t-1)\,h(t)$:

$$z(t) = Z^{1/2}(t)\,\bar{z}_\perp(t) + Q(t-1)\,h(t) \quad . \tag{28}$$

Use (28) to show that the rank-one QR-update $z(t)h^T(t)$ in (25) can be expressed as follows:

$$\begin{aligned} Q(t)R(t) = \alpha\,Q(t-1)R(t-1) &+ (1-\alpha)\,Z^{1/2}(t)\,\bar{z}_\perp(t)\,h^T(t) \\ &+ (1-\alpha)\,Q(t-1)\,h(t)h^T(t) \quad . \end{aligned} \tag{29}$$

This decomposition has the key property that it can be expressed equivalently as a product of two partitioned matrices:

$$Q(t)R(t) = \left[\begin{array}{c|c} \bar{z}_\perp(t) & Q(t-1) \end{array}\right]\left[\begin{array}{c} (1-\alpha)\,Z^{1/2}(t)\,h^T(t) \\ \hline \alpha R(t-1) + (1-\alpha)\,h(t)\,h^T(t) \end{array}\right] . \tag{30}$$

It will become apparent that this representation can be used to restore the QR structure at time t with a sequence of only $3r-3$ orthonormal Givens row rotations repesented by a multiple rotation matrix $G(t)$ where $G^T(t)G(t) = I$. For this purpose, insert the matrix product $G^T(t)G(t) = I$ between the two matrix-valued factors in (30) and split the expression into the following two recursions:

$$\left[\begin{array}{c} R(t) \\ \hline 0\cdots\cdots0 \end{array}\right] = G(t)\left[\begin{array}{c} (1-\alpha)\,Z^{1/2}(t)\,h^T(t) \\ \hline \alpha R(t-1) + (1-\alpha)\,h(t)\,h^T(t) \end{array}\right] , \tag{31a}$$

$$\left[\begin{array}{c|c} Q(t) & q_{r+1}(t) \end{array}\right] = \left[\begin{array}{c|c} \bar{z}_\perp(t) & Q(t-1) \end{array}\right] G^T(t) \quad . \tag{31b}$$

Consider recursion (31a). The Givens plane rotations in $G(t)$ must be determined so that the augmented and updated matrix $R(t-1)$ in (31a) is transformed into a strictly upper-right triangular matrix $R(t)$ by rotation. Investigate the principal nature of this problem. Although expression $\alpha R(t-1) + (1-\alpha)\,h(t)\,h^T(t)$ constitutes a "full" matrix, the update $h(t)h^T(t)$ has only rank 1 and therefore the lower partition of (31a) can be reduced to an upper Hessenberg matrix in only $r-2$ row rotations. This constitutes a first step in the transformation (31a). In a second step, the upper Hessenberg form is transformed into an upper triangular matrix using $r-1$ Hessenberg QR steps. A sequence of r Hessenberg QR steps finally produces $R(t)$. See [1] for a detailed explanation of these transformation steps. A fast $O(Nr)$ eigensubspace adaptive filter LORAF 2 can be developed on this basis. For this purpose, consider a pre-multiplication of both sides of the cross correlation time-update equation (15b) by $Q^T(t-1)$:

$$Q^T(t-1)\,c(t) = \alpha\,Q^T(t-1)\,c(t-1) + (1-\alpha)\,Q^T(t-1)z(t)y(t) \quad . \tag{32}$$

A comparison with (19a-c) reveals that (32) yields:

$$g(t) = \alpha g^*(t-1) + (1-\alpha)\,h(t)y(t) \quad . \tag{33}$$

Thus the strategy is to update $g(t)$ from the "old" $g^*(t-1)$ plus a "scalarly" weighted portion of $h(t)$. Hence only $h(t)$ must be computed explicitly from the input data $z(t)$ using data compression according to (19a) because we can show that $g^*(t)$ can be obtained from $g(t)$ by rotation. For this purpose, consider a transposed version of recursion (31b) and post-multiply both sides of this expression by $c(t)$ to obtain:

$$\left[\begin{array}{c} Q^T(t)\,c(t) \\ \hline q_{r+1}^T(t)\,c(t) \end{array}\right] = G(t)\left[\begin{array}{c} \bar{z}_\perp^T(t)\,c(t) \\ \hline Q^T(t-1)\,c(t) \end{array}\right] \quad . \tag{34}$$

Clearly, $Q^T(t)c(t) = g^*(t)$ and $Q^T(t-1)c(t) = g(t)$, therefore:

$$\begin{bmatrix} g^*(t) \\ \hline q_{r+1}^T(t)c(t) \end{bmatrix} = G(t) \begin{bmatrix} \bar{z}_\perp^T(t)\,c(t) \\ \hline g(t) \end{bmatrix} \quad . \tag{35}$$

which completes the algorithm. The recursions of the O(Nr) algorithm LORAF 2 are summarized as follows:

Initialize: $Q^T(0) = [I, 0]$; $R(0) = 0$; $c^T(0) = g^{*T}(0) = [0 \ldots 0]$

For each time step compute: (19a), (15b), (33), (26), $Z(t) = z_\perp^T(t)\,z_\perp(t)$, (27), (31a,b), (35), $p(t) = R^{-1}(t)g^*(t)$, $\hat{d}^*(t) = h^T(t)\,p(t)$, (18b).

This algorithm still assumes that $\Theta(t) = I$ according to the simplified recursion (25). In fact, experiments have shown that this is not a very limiting assumption because in practice, $\Theta(t)$ will tend to an identity matrix whenever the data characteristics changes smoothly with time and α is close to 1. Nevertheless, we continue by establishing an exact algorithm based on direct QR factor updating of the exact time update (24). For this purpose, introduce QR factors of subsequent time steps in (24) and realize that the situation is somewhat complicated at first glance because the QR structure on the right side of (24) is destroyed by $\Theta(t-1)$. Define an intermediate $r \times r$ matrix $H(t) = R(t-1)\Theta(t-1)$. Now recursion (24) can be expressed as follows:

$$Q(t)R(t) = \alpha Q(t-1)H(t) + (1-\alpha)z(t)h^T(t) \quad . \tag{36}$$

Restore the QR structure on the right side of (36) via rotation using an intermediate sequence of orthonormal Givens plane rotors $T(t)$ as follows:

$$R'(t) = T(t)H(t) \quad , \qquad Q'(t) = Q(t-1)\,T^T(t) \quad . \tag{37a,b}$$

We obtain an update equation that exhibits the desired structure:

$$Q(t)R(t) = \alpha Q'(t)R'(t) + (1-\alpha)z(t)h^T(t) \quad . \tag{38}$$

The fast direct QR updating scheme (31a,b) derived in the context of LORAF 1 is applied directly onto (38) which can be embedded in the recursions (31a,b) where only $Q \Rightarrow Q'$, $R \Rightarrow R'$ and $G \Rightarrow G'$. Fortunately, it turns out that $Q'(t)$ must not be computed explicitly. Substitute (37b) into the modified version of (31b) to realize that a direct updating scheme for Q has the following structure:

$$\begin{bmatrix} Q(t) & \vline & q_{r+1}(t) \end{bmatrix} = \begin{bmatrix} \bar{z}_\perp(t) & \vline & Q(t-1) \end{bmatrix} \begin{bmatrix} 1 & \vline & 0 \cdots 0 \\ \hline 0 & \vline & \\ \vdots & \vline & T^T(t) \\ 0 & \vline & \end{bmatrix} G'^T(t) \,. \tag{39}$$

Investigate the structure of the multiple rotation matrix $G'^T(t)$. For this purpose, pre-multiply both sides of the modified version of (31b) by $[\,\bar{z}_\perp(t) \mid Q'(t)\,]^T$. Clearly, $\bar{z}_\perp^T(t)\,\bar{z}_\perp(t) = 1$ and $Q'^T(t)Q'(t) = I$. Moreover, we must have $\bar{z}_\perp^T(t)\,Q'(t) = [0 \ldots 0]$ because $\bar{z}_\perp^T(t)\,Q(t-1) = [0 \ldots 0]$ and $Q'(t) = Q(t-1)T^T(t)$. Strictly speaking, $Q'(t)$ is just a rotated basis of the same subspace spanned by $Q(t-1)$. Thus $\bar{z}_\perp(t)$ must be orthogonal with respect to $Q'(t)$ as well. We obtain:

$$G'^T(t) = \begin{bmatrix} \bar{z}_\perp^T(t)Q(t) & \vline & \bar{z}_\perp^T(t)q_{r+1}(t) \\ \hline Q'^T(t)Q(t) & \vline & Q'^T(t)q_{r+1}(t) \end{bmatrix} \quad . \tag{40}$$

$$\begin{bmatrix} 1 & \vline & 0 \cdots 0 \\ \hline 0 & \vline & \\ \vdots & \vline & T^T(t) \\ 0 & \vline & \end{bmatrix} G'^T(t) = \begin{bmatrix} \bar{z}_\perp^T(t)Q(t) & \vline & \bar{z}_\perp^T(t)q_{r+1}(t) \\ \hline \Theta(t) & \vline & T^T(t)Q'^T(t)q_{r+1}(t) \end{bmatrix} . \tag{41}$$

Most interestingly, we have here a fairly fast scheme for computing the cosine matrix $\Theta(t)$ using only the $3r-3$ column rotations in $G'^T(t)$ applied on the augmented intermediate rotation matrix $T(t)$. According to (41), $\Theta(t)$ arises as the lower-left submatrix in this operation because $\Theta(t) = Q^T(t-1)Q(t)$ and $Q^T(t-1) = T^T(t)\,Q'^T(t)$ which follows directly from (37b). But this is not the only useful result of this operation. Continue with a substitution of (41) into (39):

$$\begin{bmatrix} Q(t) & \vline & q_{r+1}(t) \end{bmatrix} = \begin{bmatrix} \bar{z}_\perp(t) & \vline & Q(t-1) \end{bmatrix} \begin{bmatrix} \bar{z}_\perp^T(t)Q(t) & \vline & \bar{z}_\perp^T(t)q_{r+1}(t) \\ \hline \Theta(t) & \vline & T^T(t)Q'^T(t)q_{r+1}(t) \end{bmatrix}$$

Apply the rules of partitioned matrix/vector multiplication to demonstrate that:

$$Q(t) = Q(t-1)\,\Theta(t) + \bar{z}_\perp(t)\,f^T(t) \quad , \tag{42}$$

$$f^T(t) = \bar{z}_\perp^T(t)\,Q(t) \tag{43}$$

Note that $f^T(t)$ can be extracted directly from the partitoning scheme (41). Compare (42) with the initial innovations approach (22). Clearly now we realize that the rank one innovations subspace $\Delta(t)$ is given by:

$$\Delta(t) = \bar{z}_\perp(t)\,f^T(t) \quad . \tag{44}$$

This closes the circle of basic considerations about fast sequential orthogonal iteration and fast eigensubspace adaptive filters. Finally, a slight modification is necessary in the updating scheme of $g^*(t)$ originally established in (35). Incorporate $T(t)$ to show that now we must have:

$$\begin{bmatrix} g^*(t) \\ \hline q_{r+1}^T(t)c(t) \end{bmatrix} = G'(t) \begin{bmatrix} \bar{z}_\perp^T(t)c(t) \\ \hline T(t)\,g(t) \end{bmatrix} \quad . \tag{45}$$

These recursions constitute the exact fast eigensubspace or low rank adaptive filter named LORAF 3 based on a fast direct QR factor updating scheme. The recursions of this algorithm are summarized as follows:

Init.: $Q^T(0) = [I, 0]$; $R(0) = 0$; $c^T(0) = g^{*T}(0) = [0 \ldots 0]$; $\Theta(0) = I$

For each time step compute: (19a), (15b), (33), (26), $Z(t) = z_\perp^T(t)\,z_\perp(t)$, (27), $H(t) = R(t-1)\Theta(t-1)$, (31a), (41), (42), (45), $p(t) = R^{-1}(t)g^*(t)$, $\hat{d}^*(t) = h^T(t)\,p(t)$, (18b).

7. CONCLUSIONS

We have established a theory of low rank or eigensubspace adaptive filters based on sequential orthogonal iteration. The most surprising point is probably rooted in the fact that the fastest versions of these algorithms are based on a time recursive scheme for independent tracking of the QR factors of a time-varying auxiliary matrix. Experiments indicate a superior performance of the new eigensubspace adaptive filters. Experimental results are presented at the conference and in a detailed paper[1].

REFERENCES

[1] P. Strobach, "Low rank adaptive filters", IEEE Trans. on Signal Processing, submitted for publication, Aug. 1994.

[2] G.H. Golub and C.F. VanLoan, *Matrix Computations*, 2nd ed., Baltimore, MD: John Hopkins Univ. Press, 1989.

[3] P. Comon and G.H. Golub, "Tracking a few extreme singular values and vectors in signal processing", Proc. of the IEEE, Vol. 78, No. 8, pp. 1327-1343, Aug. 1990.

[4] F.L. Bauer, "Das Verfahren der Treppeniteration und verwandte Verfahren zur Lösung algebraischer Eigenwertprobleme", Z. Angew. Math. Phys., Vol. 8, pp. 214-235, 1957.