

A NOVEL CLASS OF FAST ADAPTIVE ALGORITHMS FOR MULTICHANNEL FILTERING

by

S. Theodoridis and G. Moustakides

Dept. of Computer Eng., University of Patras and C.T.I Patras.

Abstract. The major computational burden in fast transversal adaptive schemes is contributed from the time updates of the associated forward and backward predictors. In this paper the order of the predictors is assumed to be known and less or equal than that of the filter. This results in a new class of algorithms trading off performance with computational complexity.

I. Introduction. Fast least squares adaptive transversal schemes for multichannel filtering with varying number of memory taps per channel have been derived in [1] - [3]. The common feature of all these algorithms is that for each time update of the unknown filter's impulse response, time updates for the forward as well as the backward multichannel predictors are required. Furthermore the order of these predictors is implicitly assumed to be the same with the order of the filter to be identified. This is a result of the partition properties of the autocovariance matrix. However this is not necessarily true and in many cases the predictors' order is much less than the filter's order. A typical example is that of an echo canceller for the special single channel case. There, the order of the filter may be of the order of thousand, although the input signal can adequately be predicted with orders of 15-20. The aim of this paper is to overcome this apparent contradiction by incorporating into the problem the a priori information about the predictors' order. This idea was first introduced in [4] where the single channel case was treated.

The philosophy behind the proposed method is to assume that the input signal can be described by a multichannel AR model of order m_1, m_2 (AR (m_1, m_2)) with $m_1 \leq i, m_2 \leq j$, where i, j the numbers of filter taps associated with the input channels. As a consequence the Least Squares estimates of the (m_1, m_2) order multichannel predictors will be updated via an available LS time recursive scheme. Thus it is implicitly assumed that the covariance matrix of order $m_1 + m_2 + 2$ is also updated in the usual Least Squares sense. At this point we shall assume that the elements of higher order covariance matrices, can be generated from the above lower order Least Squares counterparts. This is a natural assumption since the input signal has been assumed to be AR (m_1, m_2). This way of computing the higher order covariance lags can be considered as a nonstationary generalization of a maximum entropy extension of the autocorrelation series for an AR process. Each unknown element of the covariance matrix is computed so that to maximize

the minimum prediction error power of the forward or backward predictor of an appropriate order. Exploitation of the properties of the inverse of the resulting matrix leads to an expression relating in a step up step down manner ([1] - [3]) the gain vector of order i, j with the LS predictors of order m_1, m_2 . The proof which is provided is not simply a generalization of the one given in [4] but it follows a different path revealing certain physical insights of the method.

II. Problem formulation. We shall concisely ourselves in the two channel case. The idea can obviously be extended to more general multiput multioutput cases.

We are given two input signals $x^1(n), x^2(n)$ and a desired response signal $z(n)$. We seek to compute, in anuoptimal way, the coefficients of an FIR filter

$$y(n) = - \sum_{k=0}^{i-1} c_k^1 x^1(n-k) + \sum_{k=0}^{j-1} c_k^2 x^2(n-k) \quad 2.1$$

so that $y(n)$ to follow $z(n)$ as close as possible. A time iterative solution will be given based on the stochastic Newton algorithm [5].

$$\underline{c}_{ij}(n) = \underline{c}_{ij}(n-1) + \underline{w}_{ij}(n) \underline{\xi}_{ij}(n) \quad (2.2a)$$

where

$$\underline{c}_{ij}(n) = [c_0^1(n), \dots, c_{i-1}^1(n), c_0^2(n), \dots, c_{j-1}^2(n)]^t \quad 2.2\beta$$

$$\underline{\xi}_{ij}(n) = [z(n) + \underline{c}_{ij}^t(n) \underline{x}_{ij}(n)] \quad 2.2\gamma$$

$$\underline{x}_{ij}(n) = [x^1(n) \dots x^1(n-i+1) x^2(n) \dots x^2(n-j+1)]^t \quad 2.2\delta$$

$\underline{w}_{ij}(n)$ is the gain vector defined as $\underline{w}_{ij}(n) = -R_{ij}^{-1}(n-1) \underline{x}_{ij}(n)$ with

$$R_{ij}(n) = E \left\{ \underline{x}_{ij}(n) \underline{x}_{ij}^t(n) \right\} \quad 2.2\epsilon$$

We shall assume that the input signals are

generated by a multichannel AR process of order m_1, m_2 and that estimates of the forward $\underline{A}_{m_1 m_2}(n)$ and backward $\underline{B}_{m_1 m_2}(n)$ predictors

$$\begin{bmatrix} \hat{x}_1^1(n) \\ \hat{x}_2^1(n) \end{bmatrix} = \underline{A}_{m_1 m_2}^t(n) \underline{x}_{m_1 m_2}(n) \quad 2.3a$$

$$\begin{bmatrix} \hat{x}_1^1(n - m_1) \\ \hat{x}_2^1(n - m_2) \end{bmatrix} = \underline{B}_{m_1 m_2}^t(n) \underline{x}_{m_1 m_2}(n) \quad 2.3b$$

are available for each time instant: The hat in 2.3 denotes estimates of the corresponding quantities.

III. Extension of $R_{m_1 m_2}(n)$ to $R_{ij}(n)$. This will be carried out recursively. Assume that $R_{pr}(k)$, $k=n, n-1, \dots, 0$ are known. The goal is to compute $R_{p+1 r+1}(k)$, $k=n, n-1, \dots, 0$.

We know that $R_{pr}(k) = E \left\{ \underline{x}_{pr}(k) \underline{x}_{pr}^t(k) \right\}$, $R_{p+1 r+1}(k) =$

$$E \left\{ \underline{x}_{p+1 r+1}(k) \underline{x}_{p+1 r+1}^t(k) \right\}.$$

Thus the unknown elements in $R_{p+1 r+1}(k)$ are $E\{x^1(k)x^1(k-p)\}$, $E\{x^2(k)x^2(k-r)\}$, $E\{x^1(k)x^2(k-r)\}$, $E\{x^2(k)x^1(k-p)\}$. The rest of the elements are equal with elements of $R_{pr}(k)$, $k=n, n-1, \dots$. In order to compute the four unknown elements we proceed as follows. We consider the Wiener optimum forward predictor $\underline{A}_{pr}(k)$. This is known to be given by

$$R_{pr}(k-1) \underline{A}_{pr}(k) = -E \left\{ \underline{x}_{pr}(k-1) \begin{bmatrix} x^1(k) \\ x^2(k) \end{bmatrix} \right\} \quad 3.1$$

and the corresponding minimum forward prediction error power trace

$$\text{trace} \left(E \left\{ e_{pr}^f(k) e_{pr}^{ft}(k) \right\} \right) =$$

$$\text{trace} \left(E \left\{ \begin{bmatrix} x^1(k) \\ x^2(k) \end{bmatrix} \begin{bmatrix} x^1(k), x^2(k) \end{bmatrix} \right\} \right) +$$

$$E \left\{ \begin{bmatrix} x^1(k) \\ x^2(k) \end{bmatrix} \underline{x}_{pr}^t(k-1) \right\} \underline{A}_{pr}(k) \quad 3.2$$

The unknown elements of $R_{p+1 r+1}(k)$ will be chosen so that the trace in 3.2 becomes maximum. Consider the following partition

$$\underline{x}_{pr}(k-1) = S_{pr} \begin{bmatrix} \underline{x}_{p-1 r-1}(k-1) \\ x^1(k-p) \\ x^2(k-r) \end{bmatrix} \quad 3.3$$

where S_{pr} is an appropriately defined permutation matrix. From 3.1 and 3.2 we can get

$$\underline{A}_{pr}(k) = S_{pr} \begin{bmatrix} \underline{A}_{p-1 r-1}(k) \\ \underline{O}_2 \end{bmatrix} + S_{pr} \begin{bmatrix} \underline{B}_{p-1 r-1}(k-1) \\ I_2 \end{bmatrix} \underline{k}_{pr}^f(k) \quad 3.4a$$

where $\underline{B}_{pr}(k)$ is the backward multichannel predictor.

$$R_{pr}(k) \underline{B}_{pr}(k) = -E \left\{ \underline{x}_{pr}(k) \begin{bmatrix} x^1(k-p) \\ x^2(k-r) \end{bmatrix}^t \right\} \quad 3.4b$$

and

$$\underline{K}_{pr}^f(k) = -\alpha_{pr}^b(k) \left(\underline{B}_{p-1 r-1}^t(k-1) \underline{C} + \right.$$

$$\left. E \left\{ \begin{bmatrix} x^1(k-p) \\ x^2(k-r) \end{bmatrix} \begin{bmatrix} x^1(k), x^2(k) \end{bmatrix} \right\} \right) \quad 3.4c$$

where

$$\underline{C} = E \left\{ \underline{x}_{p-1 r-1}(k-1) \begin{bmatrix} x^1(k), x^2(k) \end{bmatrix} \right\}$$

$$\alpha_{pr}^b(k) = \underline{r}_0^b(k) + \underline{B}_{pr}^t(k) E \left\{ \underline{x}_{pr}(k) \begin{bmatrix} x^1(k-p) \\ x^2(k-r) \end{bmatrix} \right\} \quad 3.4d$$

$$\underline{r}_0^b(k) = E \left\{ \begin{bmatrix} x^1(k-p) \\ x^2(k-r) \end{bmatrix} \begin{bmatrix} x^1(k-p) \\ x^2(k-r) \end{bmatrix} \right\} \quad 3.4e$$

Substituting 3.4 into 3.2 we obtain

$$\begin{aligned} \text{trace} \left(E \left\{ e_{pr}^f(k) e_{pr}^{ft}(k) \right\} \right) &= \text{trace} \left(E \left\{ e_{p-1 r-1}^f(k) e_{p-1 r-1}^{ft}(k) \right\} \right) \\ &- \text{trace} \left(\underline{k}_{pr}^{ft}(k) \alpha_{pr}^f(k) \underline{k}_{pr}^f(k) \right) \end{aligned} \quad 3.5$$

However $\alpha_{pr}^f(k)$ is a positive definite matrix thus the left hand side of 2.7 achieves its maximum value for $\underline{K}_{pr}^f(k) = 0$. Thus the unknown elements are given by

$$E \left\{ \begin{bmatrix} x^1(k-p) \\ x^2(k-r) \end{bmatrix} \begin{bmatrix} x^1(k) & x^2(k) \end{bmatrix} \right\} =$$

$$- \underline{B}_{p-1, r-1}^t(k-1) E \left\{ \underline{x}_{p-1, r-1}(k-1) \begin{bmatrix} x^1(k) & x^2(k) \end{bmatrix} \right\} \quad 3.6$$

and this is equivalent to

$$\underline{A}_{p, r}(k) = S_{p, r} \begin{bmatrix} \underline{A}_{p-1, r-1}(k) \\ O_2 \end{bmatrix} \quad k = n, n-1, \dots \quad 3.7$$

If instead of the forward predictor we consider the backward one, similar arguments will lead to

$$\underline{B}_{p, r}(k) = T_{p, r} \begin{bmatrix} O_2 \\ \underline{B}_{p-1, r-1}(k-1) \end{bmatrix} \quad k = n, n-1, \dots \quad 3.8$$

Furthermore the resulting unknown elements are identical with those resulting from the forward predictor. Following the above procedure we can obtain $R_{ij}(n)$

starting from $R_{m_1, m_2}(n)$. It is obvious from the method above that $i-m_1 = j-m_2$. This is a restriction and we cannot start from an arbitrary couple m_1, m_2 to reach i, j . The permutation matrix T_{pr} is defined from the following partitioning.

$$\underline{x}_{p, r}(k) = T_{p, r} \begin{bmatrix} x^1(k) \\ x^2(k) \\ \underline{x}_{p-1, r-1}(k) \end{bmatrix} \quad 3.9$$

The resulting covariance matrices can be shown to be positive definite.

IV. The algorithm. In this section the properties of the above extended matrix will be used to derive the novel algorithm. Define

$$\underline{A}_{p, r}(k) \equiv \begin{bmatrix} \underline{a}_{p, r}^{11}(k) & \underline{a}_{p, r}^{21}(k) \\ \underline{a}_{p, r}^{12}(k) & \underline{a}_{p, r}^{22}(k) \end{bmatrix} \quad 4.1\alpha$$

$$\underline{B}_{p, r}(k) \equiv \begin{bmatrix} \underline{b}_{p, r}^{11}(k) & \underline{b}_{p, r}^{21}(k) \\ \underline{b}_{p, r}^{12}(k) & \underline{b}_{p, r}^{22}(k) \end{bmatrix} \quad 4.1\beta$$

From the definitions in the previous section and the partition properties 3.3 and 3.9 it is shown that

$$R_{i+1, j+1}^{-1} = T_{i+1, j+1} \begin{bmatrix} 0 & \underline{O}^t \\ \underline{O} & R_{ij}^{-1}(n-1) \end{bmatrix} T_{i+1, j+1}^t +$$

$$T_{i+1, j+1} \begin{bmatrix} I_2 \\ \underline{A}_{ij}(n) \end{bmatrix} \alpha_{ij}^{-f}(n) \begin{bmatrix} I_2 & \underline{A}_{ij}^t \end{bmatrix} T_{i+1, j+1}^t \quad 4.2$$

From 4.2, 3.7, 3.4\delta and 4.1\alpha we can show that.

$$R_{i+1, j+1}^{-1}(n) =$$

$$T_{i+1, j+1} \begin{bmatrix} 0 & \underline{O}^t \\ \underline{O} & R_{ij}^{-1}(n-1) \end{bmatrix} T_{i+1, j+1}^t + \tilde{\underline{A}}(n) \alpha_{m_1, m_2}^{-f}(n) \tilde{\underline{A}}^t(n) \quad 4.3\alpha$$

$$\tilde{\underline{A}}^t(n) = \begin{bmatrix} 1 & 0 & \underline{a}_{m_1}^{11}(n) & \underline{O}_{i-m_1}^t & \underline{a}_{m_2}^{12}(n) & \underline{O}_{j-m_2}^t \\ 0 & 1 & \underline{a}_{m_1}^{21}(n) & \underline{O}_{i-m_1}^t & \underline{a}_{m_2}^{22}(n) & \underline{O}_{j-m_2}^t \end{bmatrix} T_{i+1, j+1}^t \quad 4.3\beta$$

Also using the backward predictor we obtain

$$R_{i+1, j+1}^{-1}(n) = S_{i+1, j+1} \begin{bmatrix} R_{ij}^{-1}(n) & \underline{O} \\ \underline{O}^t & 0 \end{bmatrix} S_{i+1, j+1}^t +$$

$$+ \tilde{\underline{B}}(n-i+m_1) \alpha_{m_1, m_2}^{-b}(n-i+m_1) \tilde{\underline{B}}^t(n-i+m_1) \quad 4.4\alpha$$

where

$$\tilde{\underline{B}}^t(k) \equiv$$

$$\begin{bmatrix} \underline{O}_{i-m_1}^t & \underline{b}_{m_1}^{11}(k) & \underline{O}_{j-m_2}^t & \underline{b}_{m_2}^{12}(k) & 1 & 0 \\ \underline{O}_{i-m_1}^t & \underline{b}_{m_1}^{21}(k) & \underline{O}_{j-m_2}^t & \underline{b}_{m_2}^{22}(k) & 0 & 1 \end{bmatrix} S_{i+1, j+1}^t \quad 4.4\beta$$

$\alpha_{m_1, m_2}^f(k)$ above is defined as

$$\alpha_{m_1, m_2}^f(k) = E \left\{ \begin{bmatrix} x^1(k) \\ x^2(k) \end{bmatrix} \begin{bmatrix} x^1(k) & x^2(k) \end{bmatrix} \right\} +$$

$$+ \underline{A}_{m_1, m_2}^t(k) E \left\{ \underline{x}_{m_1, m_2}(k-1) \begin{bmatrix} x^1(k) & x^2(k) \end{bmatrix} \right\} \quad 4.1\gamma$$

Combining 4.3 and 4.4 with the definition of gain, the following step up - step down recursions are obtained

$$\underline{W}_{i+1, j+1}(n) =$$

$$T_{i+1, j+1} \begin{bmatrix} 0 \\ 0 \\ \underline{w}_{ij}(n-1) \end{bmatrix} + \frac{1}{\lambda} \tilde{\underline{A}}(n-1) \alpha_{m_1, m_2}^{-f}(n-1) e_{m_1, m_2}^f(n) \quad 4.5\alpha$$

$$\underline{w}_{i+1, j+1}(n) = S_{i+1, j+1} \begin{bmatrix} \underline{w}_{ij}(n) \\ 0 \\ 0 \end{bmatrix} + \frac{1}{\lambda} \tilde{\underline{B}}(n-i+m_1-1) \alpha_{m_1, m_2}^{-b}(n-i+m_1-1) e_{m_1, m_2}^b(n-i+m_1) \quad 4.5\beta$$

where $e_{m_1, m_2}^f, e_{m_1, m_2}^b$ the a priori forward and backward errors of order m_1, m_2 .

Finally we shall obtain the relation between $\epsilon_{ij}(n)$ defined in 2.2γ and the corresponding prior error $e_{ij}(n)$. This can easily be shown to be

$$\epsilon_{ij}(n) = \frac{e_{ij}(n)}{1 - \alpha_{ij}(n)} \quad 4.6\alpha$$

where

$$\alpha_{ij}(n) = \underline{x}_{ij}^t(n) \underline{w}_{ij}(n) \quad 4.6\beta$$

From 4.4 and 4.5 we obtain the following.

$$\alpha_{ij}(n) = \alpha_{ij}(n-1) + \frac{1}{\lambda} e_{m_1, m_2}^{ft}(n) \alpha_{m_1, m_2}^{-f}(n-1) e_{m_1, m_2}^f(n) - \frac{1}{\lambda} e_{m_1, m_2}^{bt}(n+m_1-1) \alpha_{m_1, m_2}^{-b}(n+m_1-1) e_{m_1, m_2}^b(n+m_1-1) \quad 4.7$$

Combination of 2.2α, 4.5α, β, 4.5γ, 4.6 and 4.7 gives the new algorithm. The complexity is $2(i+j) + 5(m_1+m_2)$. If the input model is a true AR (m_1, m_2) then by increasing (m_1, m_2) to (i, j) there is no gain in performance. It however this is not true, then varying (m_1, m_2) from a minimum value to (i, j) a whole range of algorithms results trading off performance with computational complexity.

REFERENCES.

- [1] D. Falconer, L. Ljung: "Application of Fast Kalman Estimation To adaptive Equalization" IEEE Trans., COMM -26, pp 1439, 1446 Oct. 78.
- [2] S. Karaboyas, N. Kalouptsidis: "Efficient adaptive algorithms for ARMA identification", To appear IEEE Trans. ASSP.
- [3] S.H. Ardalan, L.J. Faber: " A fast ARMA

transversal RLS filter algorithm", IEEE Trans. ASSP - 35, pp 349 -358, March 1988.

[4] G. Moustakides, S. Theodoridis: "Fast Newton transversal filters. A New class of adaptive estimation algorithms", Submitted for publication.

[5] L. Ljung, S. Sonderstrom: "Theory and practice of Recursive identification", MIT press, 1985.