SIGNAL PROCESSING VII: Theories and Applications
M. Holt, C. Cowan, P. Grant, W. Sandham (Eds.)

A Fast Adaptive Algorithm for Multichannel System Identification - Application to DFE

Sergios THEODORIDIS, George V. MOUSTAKIDES, Kostas BERBERIDIS

Dept. of Computer Engineering, Univ. of Patras, 26500 Patras, GREECE, and
Computer Technology Institute (CTI), P.O. Box 1122, 26110 Patras, GREECE.

Abstract. The major computational contribution in fast transversal adaptive algorithms comes from updating the associated forward and backward predictors, of the input time series, which are implicitely assumed to be of the same order as the unknown system. However it is quite common in practice to extract all the predictable information of the input series with predictors of much lower order. This paper presents a method that incorporates an apriori information about the predictors' orders, assumed to be less or equal to that of the multichannel system to be identified. This results to a new class of algorithms trading off performance to computational complexity. The general multichannel problem with different number of taps per channel is treated. The applicability of the proposed algorithm to Decision Feedback Equalization is demonstrated.

1. Introduction

The design of efficient adaptive algorithms for multichannel system identification is of major importance in a wide range of disciplines such as seismic signal processing, wide band adaptive array design, digital communications, control, etc. [1]. The major computational contribution in fast transversal adaptive algorithms comes from updating the associated forward and backward predictors of the input time series, which are implicitely assumed to be of the same order as that of the unknown system [1]. However it is quite common in practice to extract all the predictable information of the input time series with predictors of much lower order. This idea was successfully exploited in the recently suggested class of Fast Newton Transversal Filters [4] where the prediction part was assumed to be of a lower order than that of the filtering part.

In general the number of taps in a multichannel FIR system need not be the same for the different channels involved (i.e. feedforward and feedback section). This is for instance the case in Decision Feedback Equalization which is the application of interest in this paper. Although such a case could be dealt by using the same number of taps for all channels, expecting the algorithm to zero the extra taps, this leads to an unnecessary computational increase and also may affect the accuracy of the obtained solution. In this paper the more general case of different number of taps per channel is treated. Simulation results verify that performance is traded off against complexity, by varying the predictor's order.

2. Formulation of the Problem

Let us assume two input signals $x^1(n)$, $x^2(n)$ which are combined by the linear system

$$\hat{d}(n) = -\sum_{i=1}^{m} c_{1i}x^1(n-i+1) - \sum_{j=1}^{l} c_{2j}x^2(n-j+1) \quad (1)$$

A recursive solution to the problem of estimating the unknown system's parameters, based on input - desired output samples is given by the well-known Stochastic Newton method [1]

$$c_{ml}(n) = c_{ml}(n-1) + w_{ml}(n)[d(n) + c_{ml}^t(n-1)x_{ml}(n)] \quad (2)$$

where $c_{ml} = [c_{11}, c_{12}, \dots c_{1m}, c_{21}, c_{22}, \dots c_{2l}]^t$ is the parameter vector and $x_{ml}(n) = [x^1(n), x^1(n-1), \dots x^1(n-m+1), x^2(n), x^2(n-1), \dots x^2(n-l+1)]^t$ The gain vector is given by $w_{ml}(n) = -\gamma(n)R_{ml}^{-1}(n)x_{ml}(n)$ where $R_{ml}(n)$ is an estimate at time n of the input data correlation matrix and $\gamma(n)$ is a properly chosen positive gain sequence. In this paper, extending the idea introduced in [4] for the single channel case, this estimate of $R_{ml}(n)$ is produced by extrapolating the sample correlation matrix of a lower order $R_{rs}(n)$, where r, s denote the prediction orders of the two input sequences respectively (with $r \leq m$ and $s \leq l$). Note that the multichannel case is not treated here as a straightforward generalization of the single channel case, having matrices in the place of scalars. The multichannel order evolution required by the algorithm, is achieved in steps involving each channel separately and leads to an algorithm involving scalar operations only.

Let us assume that the matrix $R_{rs}(n)$ is known and we seek to make an estimate of $R_{r+1s+1}(n)$. If the latter matrix is partitioned as

$$R_{r+1s+1}(n) \equiv \begin{pmatrix} K & L \\ L^t & M \end{pmatrix} \quad (3)$$

then the unknown elements in the above matrix are, the upper right element of K (denoted as $\rho_r^{11}(n)$), the upper right element of M (denoted as $\rho_s^{22}(n)$), the upper right element of L (denoted as $\tilde{\rho}_s(n)$), and the lower left element of L (denoted as $\tilde{\rho}_{-r}(n)$).

The above elements are computed from respective prediction problems following a saddle point approach. The involved prediction problems are defined as follows,

a) Given the input samples $x^1(n-1),\dots x^1(n-r)$, $x^2(n-1),\dots x^2(n-s+1)$ predict $x^2(n)$. The corresponding predictor, prediction error and error power are denoted as $A^2_{rs-1}(n)$, $e^{f2}_{rs-1}(n)$ and $\alpha^{f2}_{rs-1}(n)$ respectively.

b) Given $x^1(n-1),\dots x^1(n-r)$, $x^2(n),\dots x^2(n-s+1)$, predict $x^1(n)$. The respective prediction quantities are denoted as $\widehat{A}^1_{rs}(n)$, $\widehat{e}^{f1}_{rs}(n)$ and $\widehat{\alpha}^{f1}_{rs}(n)$.

c) Given $x^1(n),\dots x^1(n-r+1)$, $x^2(n),\dots x^2(n-s+1)$, predict $x^1(n-r)$. The respective prediction quantities are denoted as $B^1_{rs}(n)$, $e^{b1}_{rs}(n)$ and $\alpha^{b1}_{rs}(n)$.

d) Given $x^1(n),\dots x^1(n-r)$, $x^2(n),\dots x^2(n-s+1)$, predict $x^2(n-s)$. The respective prediction quantities are denoted as $B^2_{r+1s}(n)$, $e^{b2}_{r+1s}(n)$ and $\alpha^{b2}_{r+1s}(n)$.

The minmax part of the saddle point approach is equivalent with making the minimum error power to be maximum with respect to the corresponding unknown element of matrix $R_{r+1s+1}(n)$. Thus, for example, the unknown element $\tilde{\rho}_{-r}(n)$ will be estimated so as the resulting optimum predictor $A^2_{rs-1}(n)$ to have the worst possible prediction error power. Using the partitionings of Table 1 (where the definitions of the involved permutations and partitionings are given) and the well-known matrix inversion lemma for partitioned matrices [1], it can be shown after some algebra [5] that $A^2_{rs-1}(n)$ satisfies the following order update

$$A^2_{rs-1}(n) = S_{rs-1}\begin{pmatrix} A^2_{r-1s-1}(n) \\ 0 \end{pmatrix} \tag{4}$$

and the estimated value of the unknown element is given by

$$\tilde{\rho}_{-r}(n) = -B^{1t}_{r-1s-1}(n-1)\widehat{E}\{x_{r-1s-1}(n-1)x^2(n)\}$$

where $\widehat{E}\{\cdot\}$ is an estimate of $E\{\cdot\}$. As we shall see the actual computed estimates of the autocorrelation elements are of no interest to us. It is the equivalent predictor's order update which will be exploited. Similar order updates can be obtained for the rest of the predictors, that is, $\widehat{A}^1_{rs+1}(n) = \begin{pmatrix} \widehat{A}^{1t}_{rs}(n) & 0 \end{pmatrix}^t$, $B^2_{r+1s}(n) = \begin{pmatrix} 0 & \widehat{B}^{2t}_{rs}(n) \end{pmatrix}^t$, $B^1_{rs}(n) = \begin{pmatrix} 0 & \widehat{B}^{1t}_{r-1s}(n) \end{pmatrix}^t$, where $\widehat{B}^1_{r-1s}(n)$ and $\widehat{B}^2_{rs}(n)$ are intermediate auxiliary vectors computed via the relations, $\widehat{B}^1_{r-1s}(n) = T_{r-1s}\begin{pmatrix} 0 & B^{1t}_{r-1s-1}(n-1) \end{pmatrix}^t$ and $\widehat{B}^2_{r-1s}(n) = T_{rs}\begin{pmatrix} 0 & B^{2t}_{rs-1}(n-1) \end{pmatrix}^t$.

Continuing the above procedure, the matrix $R_{ml}(n)$ can be recursively estimated from $R_{rs}(n)$. Note that, with our method, this is possible only if $m-r = l-s$, imposing a restriction on the order of the matrix $R_{rs}(n)$. It can also be shown that the matrix extrapolated in the above way remains positive definite [5]. Furthermore its inverse, if viewed as a 2×2 block matrix, results in banded blocks. Specifically, in the places of the unknown elements of the extrapolated matrix, we have zeros in its inverse $R_{ml}(n)$.

3. The Multichannel FNTF Algorithm

Let us write the recursion of eq. (2) in the LS a-posteriori error formulation, i.e.

$$c_{ml}(n) = c_{ml}(n-1) + w_{ml}(n)\varepsilon_{ml}(n) \tag{5}$$

where

$$\varepsilon_{ml}(n) = d(n) + c^t_{ml}(n)x_{ml}(n) \tag{6}$$

$$w_{ml}(n) = -\lambda^{-1}R^{-1}_{ml}(n-1)x_{ml}(n) \tag{7}$$

We assume that $R_{ml}(n-1)$ is a scaled estimate of the correlation matrix extrapolated from a lower order covariance matrix $R_{r+1s+1}(n-1)$ which is computed as a least squares estimate. It is well known that the essence in deriving a fast algorithm is to achieve a fast computation of the Kalman gain vector (i.e. vector $w_{ml}(n)$ in our case). Notice that a constant weighting sequence λ has been adopted to allow for slow time variation tracking. As it has already been pointed out in the previous section the extrapolated matrices will not be explicitely involved. It is their related prediction (state space) parameters and their interrelation which will be accounted for, as it is always the case with fast algorithms.

Let us now define the partitioning $A^2_{ij}(n) \equiv \begin{pmatrix} a^{21t}_i(n) & a^{22t}_j(n) \end{pmatrix}^t$ for the predictor $A^2_{ij}(n)$ and similarly for the resting predictors $\widehat{A}^1_{ij}(n)$, $B^1_{ij}(n)$ and $B^2_{ij}(n)$. Applying successively the updating procedure as in eq. (4) and using the definitions of Table 1 we finally obtain

$$T_{ml+1}\begin{pmatrix} 1 \\ A^2_{ml}(n) \end{pmatrix} = \begin{pmatrix} a^{21}_r(n) \\ 0_{m-r} \\ 1 \\ a^{22}_s(n) \\ 0_{l-s} \end{pmatrix} \tag{8}$$

For the rest predictors it can be shown, in a similar way, that

$$\begin{pmatrix} 1 \\ \widehat{A}^1_{ml+1}(n) \end{pmatrix} = \begin{pmatrix} 1 \\ \widehat{a}^{11}_r(n) \\ 0_{m-r} \\ \widehat{a}^{12}_{s+1}(n) \\ 0_{l-s} \end{pmatrix} \tag{9}$$

$$S_{m+1l}\begin{pmatrix} B^1_{ml}(n) \\ 1 \end{pmatrix} = \begin{pmatrix} 0_{m-r} \\ b^{11}_r(n-m+r) \\ 1 \\ 0_{l-s} \\ b^{12}_s(n-l+s) \end{pmatrix} \tag{10}$$

$$B^2_{m+1l}(n) = \begin{pmatrix} 0_{m-r} \\ b^{21}_{r+1}(n-m+r) \\ 0_{l-s} \\ b^{22}_s(n-l+s) \\ 1 \end{pmatrix} \tag{11}$$

Starting from the respective definitions and using the above updating formulae the following relations are obtained for the involved errors

$$\begin{aligned} e^{f2}_{ml}(n) &= e^{f2}_{rs}(n) \\ \hat{e}^{f1}_{ml+1}(n) &= \hat{e}^{f1}_{rs+1}(n) \\ e^{b2}_{m+1l}(n) &= e^{b2}_{r+1s}(n-m+r) \\ e^{b1}_{ml}(n) &= e^{b1}_{rs}(n-m+r) \end{aligned} \quad (12)$$

and similar relations can be obtained for the involved powers. Now using (8)-(12) and following similar steps as in the two-channel staircase algorithm of [3] the corresponding two-channel FNTF algorithm of Table 2 results. It is readily observed that the computations associated with the filtering part contribute to the complexity in proportion to the systems order m, l. The contribution to the complexity of the prediction part is linearly depended on the predictors orders r, s. Specifically, the overall complexity of the algorithm of Table 2 is $10(r+s) + 2(m+l)$ MADS per time recursion, while the respective complexity of the algorithm of [3] is $12(m+l)$ MADS. Recently an exact block version of the algorithm has been developed reducing complexity to a portion of LMS per input sample [6].

4. Application to DFE

The above derived algorithm is directly applicable to Decision Feedback Equalization. To show its applicability and its performance we conducted the following experiment. A binary pseudorandom sequence was used as the bit information sequence sent to a channel which indroduced intersymbol interference. The channel was a linear phase FIR filter with an impulse response spreading over 15 successive bits. A 20dB (SNR) white Gaussian noise was added at the output of the channel. The introduced distortion was rather severe due to the large dynamic range and the deep nulls which were present in the frequency response of the channel. Equalization of channels with deep nulls suggest the use of Decision Feedback Equalizers (DFE). A typical DFE consists of the feedforward anticausal part and the feedback causal part, and its output is given by

$$\hat{x}(t) = \sum_{i=1}^{N_1} c^1_i y(t + N_1 - i) + \sum_{j=1}^{N_2} c^2_j \tilde{x}(t - j + 1) \quad (13)$$

where $\{y(t)\}$ is the received sequence and $\{\tilde{x}(t)\}$ is a sequence consisted of the correct symbols in the training mode and the detected symbols in the decision directed mode respectively. A symbol rate decision feedback equalizer is a typical two channel system identification task. The inputs in the two channels are the sequences $\{y(t)\}$ and $\{\tilde{x}(t)\}$ respectively. The equalizer parameters c^1_i and c^2_j are estimated so that the error $\hat{x}(t) - \tilde{x}(t)$ is minimized. The equalizer used by this experiment consisted of 20 feedforward and 20 feedback taps. Five curves are shown in Figure 1. Curve 1 (the lower one) corresponds to the two-channel RLS algorithm. Curve 2 (dashed line) corresponds to the MFNTF algorithm with two-channel predictors of orders 15,15. As we can see an almost negligible degradation in performance results at a computational saving of the order of 25%. Curve 3 (dotted line) corresponds to the MFNTF algorithm with predictors of orders 10,10. Curve 4 (dash-dotted line) corresponds to the MFNTF algorithm with predictors of orders 5,5. The latter has converged at about 2000 samples. In all the above cases the forgetting factor λ was taken equal to 0.99. The top curve corresponds to the Normalized LMS which at about 4000 samples (not shown in the figure) converges to the same misadjustment level as that of Curves 3 and 4. Thus we have demonstrated that the use of the multichannel Fast Newton algorithm provides the means of trading off performance with computational complexity having RLS at one end and NLMS at the other.

References

[1] N. Kalouptsidis, S. Theodoridis, (Eds.) (1993) "Adaptive System Identification and Signal Processing Algorithms", *Prentice-Hall International (UK) Limited.*

[2] F. Ling and J.G. Proakis (1985), "Adaptive Lattice Decision-Feedback Equalizers - Their Performance and Application to Time-Variant Multipath Channels", *IEEE Trans. on Communications, vol.33, no.4, pp. 348-356.*

[3] S. Karaboyas, N. Kalouptsidis (1991), "Efficient Adaptive Algorithms for ARX Identification", *IEEE Trans. on Signal Processing, vol.39, no.3, pp.571-582.*

[4] G.V. Moustakides, S. Theodoridis (1991), "Fast Newton Transversal Filters - A New Class of Adaptive Estimation Algorithms", *IEEE Trans. on Signal Processing, vol.39, no.10, pp.2184-2193.*

[5] S. Theodoridis, G.V. Moustakides, K. Berberidis (1993), "A Class of Fast Adaptive Algorithms for Multichannel System Identification and Signal Processing", *CTI Technical Report #930936.*

[6] K. Berberidis, S. Theodoridis, "A Block Adaptive Exact Fast Newton Algorithm", *CTI Technical Report #940420.*

TABLE 1: Partitions and Permutations

$$\begin{aligned} \boldsymbol{x}_{ij}(n) &= (\ x^1(n)\ \ \hat{\boldsymbol{x}}^t_{i-1j}(n)\ )^t \\ \boldsymbol{x}_{ij}(n) &= (\ \boldsymbol{x}^t_{ij-1}(n)\ \ x^2(n-j+1)\ )^t \\ \hat{\boldsymbol{x}}_{ij}(n) &= T_{ij}\ (\ x^2(n)\ \ \boldsymbol{x}^t_{ij-1}(n-1)\ )^t \\ \boldsymbol{x}_{ij}(n) &= S_{ij}\ (\ \boldsymbol{x}^t_{i-1j}(n)\ \ x^1(n-i+1)\ )^t \end{aligned}$$

TABLE 2: The MFNTF Algorithm

Define:

$$\begin{bmatrix} p_{r+1}^{11}(n) \\ p_{s+1}^{12}(n) \end{bmatrix} = \frac{1}{\lambda}\frac{\hat{e}_{rs+1}^{f1}(n)}{\hat{\alpha}_{rs+1}^{f1}(n-1)} \begin{bmatrix} 1 \\ \hat{a}_r^{11}(n-1) \\ \hat{a}_{s+1}^{12}(n-1) \end{bmatrix}, \quad \begin{bmatrix} p_{r+1}^{21}(n) \\ p_s^{22}(n) \end{bmatrix} = \frac{1}{\lambda}\frac{e_{rs}^{f2}(n)}{\alpha_{rs}^{f2}(n-1)} \begin{bmatrix} 1 \\ a_r^{21}(n-1) \\ a_s^{22}(n-1) \end{bmatrix}$$

$$\begin{bmatrix} q_{r+1}^{21}(n) \\ q_{s+1}^{22}(n) \end{bmatrix} = \frac{1}{\lambda}\frac{e_{r+1s}^{b2}(n)}{\alpha_{r+1s}^{b2}(n-1)} \begin{bmatrix} b_{r+1}^{21}(n-1) \\ b_s^{22}(n-1) \\ 1 \end{bmatrix}, \quad \begin{bmatrix} q_r^{11}(n) \\ q_{s+1}^{12}(n) \end{bmatrix} = \frac{1}{\lambda}\frac{e_{rs}^{b1}(n)}{\alpha_{rs}^{b1}(n-1)} \begin{bmatrix} b_r^{11}(n-1) \\ b_s^{12}(n-1) \\ 1 \end{bmatrix}$$

$k = n - m + r$, $\quad [p]_k$ denotes the k-th element of the vector p

- Available from the previous recursion of the MFNTF: $w_{ml}(n-1)$, $\alpha_{ml}(n-1)$
- Available from any LS multichannel algorithm: $\hat{e}_{rs+1}^{f1}(n)$, $e_{rs}^{f2}(n)$, $e_{rs}^{b1}(k)$, $e_{r+1s}^{b2}(k)$
  $p_{r+1}^{11}(n)$, $p_{s+1}^{12}(n)$, $p_{r+1}^{21}(n)$, $p_s^{22}(n)$, $q_r^{11}(k)$, $q_{s+1}^{12}(k)$, $q_{r+1}^{21}(k)$, $q_{s+1}^{22}(k)$

Prediction Part

$$\hat{w}_{ml+1}(n) = T_{ml+1}\,[\;0 \quad w_{ml}^t(n-1)\;]^t - T_{ml+1}\,[\;p_{r+1}^{21t}(n) \quad 0_{m-r}^t \quad p_s^{22t}(n) \quad 0_{l-s}^t\;]^t$$

$$w_{m+1l+1}(n) = [\;0 \quad \hat{w}_{ml+1}^t(n)\;]^t - [\;p_{r+1}^{11t}(n) \quad 0_{m-r}^t \quad p_{s+1}^{12t}(n) \quad 0_{l-s}^t\;]^t$$

$$[\;w_{m+1l}^t(n) \quad 0\;]^t = w_{m+1l+1}(n) + [\;0_{m-r}^t \quad q_{r+1}^{21t}(k) \quad 0_{l-s}^t \quad q_{s+1}^{22t}(k)\;]^t$$

$$[\;w_{ml}^t(n) \quad 0\;]^t = S_{m+1l}^t w_{m+1l}(n) + [\;0_{m-r}^t \quad q_r^{11t}(k) \quad 0_{l-s}^t \quad q_{s+1}^{22t}(k)\;]^t$$

$$\alpha_{ml}(n) = \alpha_{ml}(n-1) + [p_{r+1}^{11}]_1 \hat{e}_{rs+1}^{f1}(n) + [p_{r+1}^{21}]_1 e_{rs}^{f2}(n) - [q_{s+1}^{22}]_{s+1} e_{r+1s}^{b2}(k) - [q_{s+1}^{12}]_{s+1} e_{rs}^{b1}(k)$$

Filtering Part

$$e_{ml}(n) = d(n) + c_{ml}^t(n-1)x_{ml}(n)$$

$$\varepsilon_{ml}(n) = e_{ml}(n)/\alpha_{ml}(n)$$

$$c_{ml}(n) = c_{ml}(n-1) + w_{ml}(n)\varepsilon_{ml}(n)$$

FIGURE 1