neither W^* nor α^* would be computed more than once. In addition, the complexity of $O(K2^{K+1}r^2)$ MAD for the overall backward recursion algorithm can be obtained. Please note that in the course of performing the time backward update recursions from 80 to 79 for all possible subset autoregressions, only the parameter matrices and the necessary W^* and α^* at 80 are required to acquire all parameter matrices and both W^* and α^* at 79. No parameters at $80+i, i > 1$ are required, and neither W^* nor α^* at $80+i$ is needed.

In addition, note that the initialization of both forward and backward time update recursions can be carried out by the direct method for a middle block of the sample set; then, one processor may perform forward time update recursions, and the other processor may perform backward time update recursions concurrently. This method has been used in estimating a gradually changing spectrum for the Canadian lynx [13] and is useful in signal smoothing. In addition, this structure provides great benefit in working with recursive algorithms suitable for carrying out within a multi-c.p.u. computing environment.

Further, an order selection criterion, such as the BIC, can be used at each time instant with the proposed method to select the optimum subset AR. For further details, the interested reader is referred to [13], which is available upon request. In addition, it is unnecessary to neglect possible zero constraints in the nonzero coefficient matrices of the optimum subset AR model selected. Our previous experience [8] shows that the proposed search algorithm in conjunction with model selection criteria can, with slight modification, select the optimum subset AR with zero constraints using the prewindowed case.

IV. Conclusion

To this point, we have developed forward and backward time update recursions for the prewindowed case that recursively estimate the parameter matrices for a multichannel subset autoregression. The algorithm is computationally efficient, avoids cumbersome matrix inversion, and provides the obvious relations to link multichannel subset autoregressions at consecutive time instants.

References

[1] F. Ling and J. G. Proakis, "A generalised multichannel least squares lattice algorithm based on sequential processing stages," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 381–389, Apr. 1984.

[2] D. W. Lin, "On digital implementation of the fast Kalman algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 998–1005, 1984.

[3] D. T. M. Slock and T. Kailath, "A modular multichannel multi-experiment fast transversal filter and a prewindowed framework for covariance RLS algorithm," *Signal Processing*, vol. 28, pp. 46–61, 1992.

[4] H. Lev-ari, T. Kailath, and J. Cioffi, "Least-squares adaptive lattice and transversal filters: A unified geometric theory," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 222–235, Mar. 1984.

[5] S. M. Kay and S. L. Marple, "Spectrum analysis—A modern perspective," *Proc. IEEE*, vol. 69, pp. 1380–1419, Nov. 1981.

[6] C. Carayannis, D. Manolakis, and N. Kalouptsidis, "A unified view of parametric processing algorithms for prewindowed signals," *Signal Processing*, vol. 10, pp. 335–368, June 1986.

[7] J. H. W. Penm and R. D. Terrell, "A note on the recursions of multichannel complex subset autoregressions," *IEEE Trans. Automatic Contr.*, vol. AC-28, no. 4, pp. 534–536, 1983.

[8] ——, "Multivariate subset autoregressive modelling with zero constraints for detecting 'Overall Causality', " *J. Econometrics*, vol. 24, pp. 311–330, 1984.

[9] J. H. W. Penm, J. H. C. Penm, and R. D. Terrell, "Using the bootstrap as an aid in choosing the approximate representation for vector time series," *J. Business & Econ. Stat.*, vol. 10, no. 2, pp. 213–219, Apr. 1992.

[10] ——, " The recursive fitting of subset VARX models," forthcoming in *J. Time Series Anal.*, vo. 14, no. 6, pp. 603–619, Nov. 1993.

[11] B. Friedlander and B. Porat, "Multichannel ARMA spectral estimation by the modified Yule-Walker method," *Signal Processing* , vol 10, no. 1, pp. 49–59, 1986.

[12] H. Sakai, "An application of a BIC-type method to harmonic analysis and a new criterion for order determination of an AR process," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 6, pp. 999–1004, June 1990.

[13] J. H. W. Penm, J. H. C. Penm, and R. D. Terrell, "On the sequential fitting of subset autoregressions using the prewindowed case," *Memo*, Australian Nat. Univ., Oct., 1992.

A Fast Newton Multichannel Algorithm for Decision Feedback Equalization

Sergios Theodoridis, George V. Moustakides, and Kostas Berberidis

***Abstract*—Fast Newton transversal filters (FNTF) are a new class of adaptive algorithms unifying NLMS and fast RLS. This is achieved by allowing the algorithm to exploit *a priori* information about the associated predictors order. This note extends FNTF to the multichannel case and demonstrates the applicability of the resulting algorithm to decision feedback equalization.**

I. Introduction

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], [2]. 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 (FNTF) [7] where the prediction part was assumed to be of lower order than that of the filtering part.

In this note FNTF is extended to the multichannel case so as to be directly applicable to decision feedback equalization. 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.

II. Problem Formulation and Correlation Matrix Extrapolation

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

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

Manuscript received January 30, 1993; revised June 11, 1994. The associate editor coordinating the review of this paper and approving it for publication was Dr. Fuyun Ling.

The authors are with the Department of Computer Engineering and Informatics, University of Patras, Rio 26500, Greece, and with the Computer Technology Institute (CTI), Patras 26110, Greece.

IEEE Log Number 9406838.

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]

$$\mathbf{c}_{ml}(n) = \mathbf{c}_{ml}(n-1) + \mathbf{w}_{ml}(n)[d(n) + \mathbf{c}'_{ml}(n-1)\mathbf{x}_{ml}(n)] \quad (2)$$

where $\mathbf{c}_{ml} = [c_{11}, c_{12}, \ldots c_{1m}, c_{21}, c_{22}, \ldots c_{2l}]'$ is the parameter vector and

$$\mathbf{x}_{ml}(n) = [x^1(n), x^1(n-1), \ldots x^1(n-m+1), x^2(n), x^2(n-1), \ldots x^2(n-l+1)]'. \quad (3)$$

The gain vector is given by $\mathbf{w}_{ml}(n) = -\gamma(n)R^{-1}_{ml}(n)\mathbf{x}_{ml}(n)$ 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 correspondence, extending the idea introduced in [7] 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 $r \le m$ and $s \le l$). Note that 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

$$R_{r+1s+1}(n) \equiv \begin{pmatrix} K & L \\ L' & M \end{pmatrix} \quad (4)$$

then the unknown elements in the above matrix are, the upper right element of K (denoted as $\rho^{11}_r(n)$), the upper right element of M (denoted as $\rho^{22}_s(n)$), 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 $x^1(n-1), \ldots x^1(n-r), x^2(n-1), \ldots x^2(n-s+1)$ 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 $x^1(n-1), \ldots x^1(n-r), x^2(n), \ldots x^2(n-s+1)$, predict $x^1(n)$. The respective prediction quantities are denoted as $\hat{A}^1_{rs}(n)$, $\hat{e}^{f1}_{rs}(n)$ and $\hat{\alpha}^{f1}_{rs}(n)$.

c) Given $x^1(n), \ldots x^1(n-r+1), x^2(n), \ldots x^2(n-s+1)$, 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 $x^1(n), \ldots x^1(n-r), x^2(n), \ldots x^2(n-s+1)$, 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 the matrix $R_{r+1s+1}(n)$. For example, consider the unknown element $\tilde{\rho}_{-r}(n)$ and the respective optimum predictor $A^2_{rs-1}(n)$. Then both the predictor and the resulting minimum error power are functions of $\tilde{\rho}_{-r}(n)$. Our goal will be to select the unknown element so as to maximize this minimum error power. Using the partitionings of Table I (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 that

$$A^2_{rs-1}(n) = S_{rs-1}\begin{pmatrix} A^2_{r-1s-1}(n) \\ 0 \end{pmatrix} + S_{rs-1}\begin{pmatrix} B^1_{r-1s-1}(n-1) \\ 1 \end{pmatrix} k^{\alpha 2}_{rs-1} \quad (5)$$

TABLE I
THE INVOLVED PARTITIONINGS AND PERMUTATIONS

$x_{ij}(n) = \begin{pmatrix} x^1(n) \\ \hat{x}_{i-1j}(n) \end{pmatrix}$	$x_{ij}(n) = \begin{pmatrix} x_{ij-1}(n) \\ x^2(n-j+1) \end{pmatrix}$
$\hat{x}_{ij}(n) = T_{ij}\begin{pmatrix} x^2(n) \\ x_{ij-1}(n-1) \end{pmatrix}$	$x_{ij}(n) = S_{ij}\begin{pmatrix} x_{i-1j}(n) \\ x^1(n-i+1) \end{pmatrix}$

with

$$k^{\alpha 2}_{rs-1} = -\alpha^{-b1}_{r-1s-1}(n-1)[B^{1t}_{r-1s-1}(n-1) \times \hat{E}\{\mathbf{x}_{r-1s-1}(n-1)x^2(n)\} + \tilde{\rho}_{-r}(n)]$$

where $\hat{E}\{\cdot\}$ is an estimate of the expectation $E\{\cdot\}$. Substituting the above two equations into the definition of the error power $\alpha^{f2}_{rs-1}(n)$ results in

$$\alpha^{f2}_{rs-1}(n) = \alpha^{f2}_{r-1s-1}(n) - \alpha^{-b1}_{r-1s-1}(n-1)(k^{\alpha 2}_{rs-1})^2. \quad (6)$$

From (6) it is apparent that $\alpha^{f2}_{rs-1}(n)$ achieves its maximum value if the reflection coefficient $k^{\alpha 2}_{rs-1} = 0$, which is equivalent to the following order update:

$$A^2_{rs-1}(n) = S_{rs-1}\begin{pmatrix} A^2_{r-1s-1}(n) \\ 0 \end{pmatrix}. \quad (7)$$

Note that each unknown element can be estimated by more than one minmax problem. However, the resulting element is always the same [8]. It can be easily shown that the rest of the predictors are order-updated in a similar way as above, that is, $\hat{A}^1_{rs+1}(n) = (\hat{A}^{1t}_{rs}(n) \quad 0)'$, $B^2_{r+1s}(n) = (0 \quad \hat{B}^{2t}_{rs}(n))^t$, $B^1_{rs}(n) = (0 \quad \hat{B}^{1t}_{r-1s}(n))'$, where $\hat{B}^1_{r-1s}(n)$ and $\hat{B}^2_{rs}(n)$ are intermediate auxiliary vectors computed via the relations $\hat{B}^1_{r-1s}(n) = T_{r-1s}(0 \quad B^{1t}_{r-1s-1}(n-1))'$ and $\hat{B}^2_{r-1s}(n) = T_{rs}(0 \quad B^{2t}_{rs-1}(n-1))'$.

Continuing the above procedure, the matrix $R_{ml}(n)$ is 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 [8]. 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.

III. Derivation of the Algorithm

Our starting point is the recursion (2) written in the least squares posterior error formulation using the dual Kalman gain [6] that is

$$\mathbf{c}_{ml}(n) = \mathbf{c}_{ml}(n-1) + \mathbf{w}_{ml}(n)\varepsilon_{ml}(n) \quad (8)$$

where

$$\varepsilon_{ml}(n) = d(n) + \mathbf{c}'_{ml}(n)\mathbf{x}_{ml}(n) \quad (9)$$

$$\mathbf{w}_{ml}(n) = -\lambda^{-1}R^{-1}_{ml}(n-1)\mathbf{x}_{ml}(n). \quad (10)$$

It is already well established that the essence in deriving a fast algorithm is the fast computation of the Kalman gain vector. In this section 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. That is

$$R_{r+1s+1}(n) = \lambda R_{r+1s+1}(n-1) + \mathbf{x}_{r+1s+1}(n)\mathbf{x}'_{r+1s+1}(n). \quad (11)$$

TABLE II
THE MULTICHANNEL FAST NEWTON TRANSVERSAL FILTERING (MFNTF) ALGORITHM

Define:

$$\begin{bmatrix} \boldsymbol{p}^{11}_{r+1}(n) \\ \boldsymbol{p}^{12}_{s+1}(n) \end{bmatrix} = \frac{1}{\lambda}\frac{\widehat{e}^{f1}_{rs+1}(n)}{\widehat{\alpha}^{f1}_{rs+1}(n-1)} \begin{bmatrix} 1 \\ \widehat{\boldsymbol{a}}^{11}_{r}(n-1) \\ \widehat{\boldsymbol{a}}^{12}_{s+1}(n-1) \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{p}^{21}_{r+1}(n) \\ \boldsymbol{p}^{22}_{s}(n) \end{bmatrix} = \frac{1}{\lambda}\frac{e^{f2}_{rs}(n)}{\alpha^{f2}_{rs}(n-1)} \begin{bmatrix} 1 \\ \boldsymbol{a}^{21}_{r}(n-1) \\ \boldsymbol{a}^{22}_{s}(n-1) \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{q}^{21}_{r+1}(n) \\ \boldsymbol{q}^{22}_{s+1}(n) \end{bmatrix} = \frac{1}{\lambda}\frac{e^{b2}_{r+1s}(n)}{\alpha^{b2}_{r+1s}(n-1)} \begin{bmatrix} \boldsymbol{b}^{21}_{r+1}(n-1) \\ \boldsymbol{b}^{22}_{s}(n-1) \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{q}^{11}_{r}(n) \\ \boldsymbol{q}^{12}_{s+1}(n) \end{bmatrix} = \frac{1}{\lambda}\frac{e^{b1}_{rs}(n)}{\alpha^{b1}_{rs}(n-1)} \begin{bmatrix} \boldsymbol{b}^{11}_{r}(n-1) \\ \boldsymbol{b}^{12}_{s}(n-1) \\ 1 \end{bmatrix}$$

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

- Available from the previous recursion of the MFNTF: $\boldsymbol{w}_{ml}(n-1)$, $\alpha_{ml}(n-1)$
- Available from any LS multichannel algorithm: $\widehat{e}^{f1}_{rs+1}(n)$, $e^{f2}_{rs}(n)$, $e^{b1}_{rs}(k)$, $e^{b2}_{r+1s}(k)$
  $\boldsymbol{p}^{11}_{r+1}(n)$, $\boldsymbol{p}^{12}_{s+1}(n)$, $\boldsymbol{p}^{21}_{r+1}(n)$, $\boldsymbol{p}^{22}_{s}(n)$, $\boldsymbol{q}^{11}_{r}(k)$, $\boldsymbol{q}^{12}_{s+1}(k)$, $\boldsymbol{q}^{21}_{r+1}(k)$, $\boldsymbol{q}^{22}_{s+1}(k)$

Prediction Part

$$\widehat{\boldsymbol{w}}_{ml+1}(n) = T_{ml+1}\begin{bmatrix} 0 \\ \boldsymbol{w}_{ml}(n-1) \end{bmatrix} - T_{ml+1}\begin{bmatrix} \boldsymbol{p}^{21}_{r+1}(n) \\ \mathbf{0}_{m-r} \\ \boldsymbol{p}^{22}_{s}(n) \\ \mathbf{0}_{l-s} \end{bmatrix}$$

$$\boldsymbol{w}_{m+1l+1}(n) = \begin{bmatrix} 0 \\ \widehat{\boldsymbol{w}}_{ml+1}(n) \end{bmatrix} - \begin{bmatrix} \boldsymbol{p}^{11}_{r+1}(n) \\ \mathbf{0}_{m-r} \\ \boldsymbol{p}^{12}_{s+1}(n) \\ \mathbf{0}_{l-s} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{w}_{m+1l}(n) \\ 0 \end{bmatrix} = \boldsymbol{w}_{m+1l+1}(n) + \begin{bmatrix} \mathbf{0}_{m-r} \\ \boldsymbol{q}^{21}_{r+1}(k) \\ \mathbf{0}_{l-s} \\ \boldsymbol{q}^{22}_{s+1}(k) \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{w}_{ml}(n) \\ 0 \end{bmatrix} = S^{t}_{m+1l}\boldsymbol{w}_{m+1l}(n) + \begin{bmatrix} \mathbf{0}_{m-r} \\ \boldsymbol{q}^{11}_{r}(k) \\ \mathbf{0}_{l-s} \\ \boldsymbol{q}^{22}_{s+1}(k) \end{bmatrix}$$

$$\alpha_{ml}(n) = \alpha_{ml}(n-1) + [\boldsymbol{p}^{11}_{r+1}]_1\widehat{e}^{f1}_{rs+1}(n) + [\boldsymbol{p}^{21}_{r+1}]_1 e^{f2}_{rs}(n) - [\boldsymbol{q}^{22}_{s+1}]_{s+1}e^{b2}_{r+1s}(k) - [\boldsymbol{q}^{12}_{s+1}]_{s+1}e^{b1}_{rs}(k)$$

Filtering Part

$$e_{ml}(n) = d(n) + \boldsymbol{c}^{t}_{ml}(n-1)\boldsymbol{x}_{ml}(n)$$
$$\varepsilon_{ml}(n) = e_{ml}(n)/\alpha_{ml}(n)$$
$$\boldsymbol{c}_{ml}(n) = \boldsymbol{c}_{ml}(n-1) + \boldsymbol{w}_{ml}(n)\varepsilon_{ml}(n)$$

A constant weighting sequence λ has been adopted to allow for slow time variation tracking. As has already been pointed out in the previous section, these matrices will not be explicitely involved. It is their related prediction (state space) parameters and their interrelation which will be accounted for, as is always the case with fast algorithms. Let us now define the partitioning $A^{2}_{ij}(n) \equiv (\mathbf{a}^{21t}_{i}(n) \;\; \mathbf{a}^{22t}_{j}(n))^t$ for the predictor $A^{2}_{ij}(n)$ and similarly for the remaining predictors $\hat{A}^{1}_{ij}(n)$, $B^{1}_{ij}(n)$ and $B^{2}_{ij}(n)$. Applying successively the updating procedure as in (7) and using the definitions of Table I, we finally obtain

$$T_{ml+1}(1 \;\; A^{2t}_{ml}(n))^t = (\mathbf{a}^{21t}_{r}(n) \;\; \mathbf{0}^{t}_{m-r} \;\; 1 \;\; \mathbf{a}^{22t}_{s}(n) \;\; \mathbf{0}^{t}_{l-s})^t. \tag{12}$$

In a similar way it can be shown that

$$(1 \;\; \hat{A}^{1t}_{ml+1}(n))^t = (1 \;\; \hat{\mathbf{a}}^{11t}_{r}(n) \;\; \mathbf{0}^{t}_{m-r} \;\; \hat{\mathbf{a}}^{12t}_{s+1}(n) \;\; \mathbf{0}^{t}_{l-s})^t \tag{13}$$

$$S_{m+1l}(B^{1t}_{ml}(n) \;\; 1)^t = (\mathbf{0}^{t}_{m-r} \;\; \mathbf{b}^{11t}_{r}(n-m+r) \;\; 1 \;\; \mathbf{0}^{t}_{l-s} \;\; \mathbf{b}^{12t}_{s}(n-l+s))^t \tag{14}$$

$$B^{2}_{m+1l}(n) = (\mathbf{0}^{t}_{m-r} \;\; \mathbf{b}^{21t}_{r+1}(n-m+r) \;\; \mathbf{0}^{t}_{l-s} \;\; \mathbf{b}^{22t}_{s}(n-l+s) \;\; 1)^t. \tag{15}$$

Using the above and the respective definitions, the following relationships can be obtained for the involved errors and error power variables:

$$\begin{aligned}
e^{f2}_{ml}(n) &= e^{f2}_{rs}(n) & \alpha^{f2}_{ml}(n-1) &= \alpha^{f2}_{rs}(n-1)\\
\hat{e}^{f1}_{ml+1}(n) &= \hat{e}^{f1}_{rs+1}(n) & \hat{\alpha}^{f1}_{ml+1}(n-1) &= \hat{\alpha}^{f1}_{rs+1}(n-1)\\
e^{b2}_{m+1l}(n) &= e^{b2}_{r+1s}(n-m+r) & \alpha^{b2}_{m+1l}(n-1) &= \alpha^{b2}_{r+1s}(n-m+r-1)\\
e^{b1}_{ml}(n) &= e^{b1}_{rs}(n-m+r) & \alpha^{b1}_{ml}(n-1) &= \alpha^{b1}_{rs}(n-m+r-1).
\end{aligned} \tag{16}$$

Introducing (12)–(16) in the two-channel staircase algorithm of [6], the corresponding two-channel FNTF algorithm of Table II 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

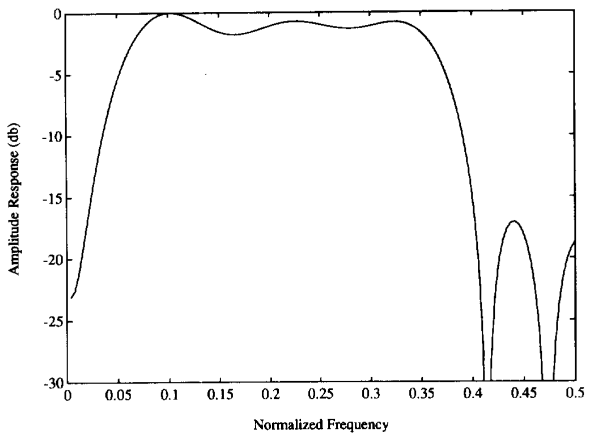

Fig. 1. Amplitude response in dB versus normalized frequency of the channel used in the adaptive equalization experiment.

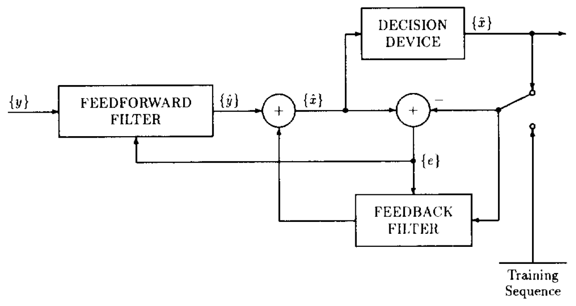

Fig. 2. Block diagram of the basic decision feedback equalizer structure.

linearly dependent on the predictors orders r, s. Specifically, the overall complexity of the algorithm of Table II is $10\ (r + s) + 2\ (m + 1)$ MADS per time recursion, while the respective complexity of the algorithm of [6] is $12(m + l)$ MADS.

IV. Application to Decision Feedback Equalization

To verify the above theoretical developments, a channel equalization experiment was carried out. A binary pseudorandom sequence was used as the bit information sequence sent to a channel which introduces intersymbol interference. An FIR linear phase channel was used with an impulse response spreading over 15 successive bits. A 20 dB (SNR) white Gaussian noise was added at the output of the channel. Fig. 1 shows the frequency response of the channel. The distortion which is introduced is rather severe, due to the large dynamic range and the deep nulls which are present in the frequency response. Equalization of channels with deep nulls suggest the use of decision feedback equalizers (DFE). Fig. 2 shows the typical structure of a DFE. It consists of the feedforward anticausal part and the feedback causal part, and it is described as

$$\hat{x}(t) = \sum_{i=1}^{N_1} c_i^1 y(t + N_1 - i) + \sum_{j=1}^{N_2} c_j^2 \tilde{x}(t - j + 1) \qquad (17)$$

where $\{y(t)\}$ is the received sequence. In the training mode $\tilde{x}(t)$ are the correct symbols and in the decision-directed mode the detected

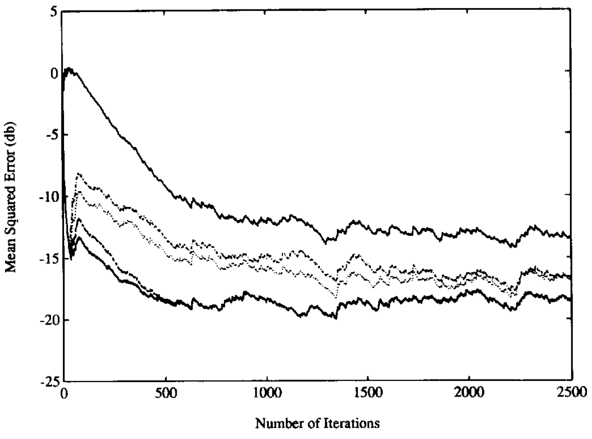

Fig. 3. MSE versus time for the DFE experiment: (a) Bottom solid line—two-channel RLS with predictor orders 20, 20; (b) dashed line—MFNTF with predictor orders 15, 15; (c) Dotted line—MFNTF with predictor orders 10, 10; (d) Dash-dotted line—MFNTF with predictor orders 5, 5; (e) top solid line—normalized LMS with $\mu = 0.25$.

symbols. 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_i^1 and c_j^2 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 Fig. 3. 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 (dashdotted 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.

V. Conclusion

This correspondence has proposed a new structure for adaptive multichannel filtering. The resulting algorithm trades off complexity with performance covering the whole range between multichannel RLS and multichannel LMS. The applicability of the algorithm to decision feedback equalization has successfully been demonstrated by simulations.

References

[1] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[2] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1983.
[3] F. Ling and J. G. Proakis, "Adaptive lattice decision-feedback equalizers—Their performance and application to time-variant multipath channels," *IEEE Trans. Commun.*, vol. COM-33, no. 4, pp. 348–356, Apr. 1985.

[4] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Mag.*, pp. 10–26, July 1991.
[5] E. Karlsson, "Adaptive polynomial lattice filters," in *Proc. ICASSP-91* (Toronto), Apr. 1991.
[6] S. Karaboyas and N. Kalouptsidis, "Efficient adaptive algorithms for ARX identification," *IEEE Trans. Signal Processing*, vol. 39, no. 3, pp. 571–582, Mar. 1991.
[7] G. V. Moustakides and S. Theodoridis, "Fast Newton transversal filters—A new class of adaptive estimation algorithms," *IEEE Trans. Signal Processing*, vol. 39, no. 10, pp. 2184–2193, Oct. 1991.
[8] S. Theodoridis, G. V. Moustakides, and K. Berberidis, "A class of fast adaptive algorithms for multichannel system identification and signal processing," *CTI Tech. Rep. #930936*, Sept. 1993.

An Efficient CORDIC Array Structure for the Implementation of Discrete Cosine Transform

Yu Hen Hu and Zhenyang Wu

***Abstract*—We propose a novel implementation of the discrete cosine transform (DCT) and the inverse DCT (IDCT) algorithms using a CORDIC (COordinate Rotation DIgital Computer)-based systolic processor array structure. First, we reformulate an N-point DCT or IDCT algorithm into a rotation formulation which makes it suitable for CORDIC processor implementation. We then propose to use a pipelined CORDIC processor as the basic building block to construct 1-D and 2-D systolic-type processor arrays to speed up the DCT and IDCT computation. Due to the proposed novel rotation formulation, we achieve 100% processor utilization in both 1-D and 2-D configurations. Furthermore, we show that for the 2-D configurations, the same data processing throughput rate can be maintained as long as the processor array dimensions are increased linearly with N. Neither the algorithm formulation or the array configuration need to be modified. Hence, the proposed parallel architecture is *scalable* to the problem size. These desirable features make this novel implementation compare favorably to previously proposed DCT implementations.**

I. Introduction

In this correspondence, we present an efficient implementation of the discrete cosine transform (DCT) algorithm [1] and its inverse (IDCT) using a CORDIC processor array structure. DCT has been incorporated into image compression standards such as JPEG, MPEG, and CCITT H261. It has also found many applications in speech coding and realization of filter banks for frequency-division and time-division multiplexer (FDM-TDM) systems. Due to its increasing importance, numerous attempts have been made to accelerate the DCT computation in order to facilitate real time, high-throughput implementation [2]. One family of approaches is to derive fast DCT algorithms [7]–[12] by reducing the number of multiplications needed in the formulation. Yet another family of approaches [3]–[6] has focused on using hardware implementation of DCT with parallel or pipelined VLSI array structures [13]. While most of these proposed implementations are based on the multiply-and-add-type arithmetic units, some [5] have reported implementations using a special arithmetic unit called CORDIC.

CORDIC (COordinate Rotation DIgital Computer) is a rotation-based arithmetic algorithm which is particularly efficient for the evaluation of fast transformation algorithms such as DFT (discrete Fourier transform), FFT (fast Fourier transform) [15], and DHT (discrete Hartly transform) [16]. In this correspondence, we will propose new formulations of both the DCT and the IDCT algorithms to facilitate very efficient implementation using CORDIC processor array structures. Compared to the previous result [5], our implementations require only local data communication, have simple, regular array structures, and are linearly scalable.

II. Vector Rotation Formulation of DCT and IDCT Algorithm

Given a real-valued sequence $\{x(n);\ 0 \le n \le N-1\}$, the DCT of $\{x(n)\}$ is defined by

$$X(k) = \frac{2}{N}c(k)\sum_{n=0}^{N-1} x(n)\cos\left[\frac{\pi(2n+1)k}{2N}\right]$$
$$0 \le k \le N-1 \quad (1)$$

and the IDCT of an N-point real-valued sequence $\{X(k); 0 \le k \le N-1\}$ is defined by

$$x(n) = \sum_{k=0}^{N-1} c(k)X(k)\cos\left[\frac{\pi(2n+1)k}{2N}\right], \quad 0 \le n \le N-1 \quad (2)$$

where $c(0) = \frac{1}{\sqrt{2}}$, and $c(k) = 1$ for $1 \le k \le N-1$. Since $\frac{2}{N}$ is a scaling factor which can easily be computed if N is a power of 2, we need only to compute $\tilde{X}(k) = NX(k)/2$. Let us define

$$V(k) = \sum_{n=0}^{N-1} x(n)\exp\left[j\frac{\pi(2n+1)k}{2N}\right], \quad 0 \le k \le N-1. \quad (3)$$

Clearly, $\tilde{X}(k) = \mathrm{Re}\{V(k)\}$ for $k \ge 1$, and $\tilde{X}(0) = \frac{1}{\sqrt{2}}V(0)$. Assuming that N is an even number, our strategy is to decompose $V(k)$ such that

$$V(k) = V_e(k) + V_0(k) \quad (4)$$

where

$$V_e(k) = \sum_{n=0}^{N/2-1} x(2n)\exp\left[j\frac{\pi(4n+1)k}{2N}\right] \quad (5)$$

and

$$V_0(k) = \sum_{m=0}^{N/2-1} x(2m+1)\exp\left[j\frac{\pi(4m+3)k}{2N}\right]. \quad (6)$$

The following relations can be verified easily: $\mathrm{Re}\{V_e(N-k)\} = \mathrm{Im}\{V_e(k)\}$ and $\mathrm{Re}\{V_0(N-k)\} = -\mathrm{Im}\{V_0(k)\}$. Substitute $m =$

Manuscript received March 10, 1993; revised May 26, 1994. The associate editor coordinating the review of this paper and approving it for publication was Prof. Daniel Fuhrmann.

Y. H. Hu is with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53706 USA.

Z. Wu is with the Department of Radio Engineering, Southeast University, Nanjing 210018, People's Republic of China.

IEEE Log Number 9406835.