D5.5

A New 2-D Fast RLS Algorithm

A. M. Sequeira and C. W. Therrien*
Naval Postgraduate School
Monterey, California 93943

Abstract

A two-dimensional fast recursive least squares algorithm is presented using a geometrical formulation based on the mathematical concepts of vector space, orthogonal projection, and subspace decomposition.

By appropriately ordering the 2-D data, the algorithm provides an exact least-squares solution to the deterministic Normal equations. The method is further extended to the general FIR Wiener filter and to ARMA modeling. The size and shape of the support region for both the MA and AR coefficients of the filter can be chosen arbitrarily.

1 Introduction

Adaptive algorithms have been used successfully in a wide range of signal processing applications involving unknown or non-stationary data. Real time implementation of these algorithms has recently become possible with the latest VLSI technology partly as a result of efficient computational algorithms such as the 'fast' recursive least squares (fast RLS). The fast RLS methods are based upon an elegant geometric approach involving the concepts of linear vector spaces, orthogonality, projection operators, and their relation to linear least squares prediction [1,2,3]. The development of adaptive algorithms for two-dimensional (2-D) problems has been much slower than their development for one-dimensional (1-D) problems. In this paper we describe a fast RLS method for 2-D signals. We show that besides the 2-D filter, the method involves the use and simultaneous update of two additional multichannel filters, and related gain transversal filters. In this way the algorithm is similar but more extensive than the 1-D counterpart.

*Sr. Member of the IEEE.

2 2-D RLS Algorithm

2.1 2-D Adaptive Filtering

In the following we consider a 2-D linear predictive filter of the form

$$\hat{y}(n_1,n_2) = \sum_{i=0}^{N}\sum_{j=0}^{M} a_{ij}\, y(n_1-i, n_2-j) \qquad (i,j) \neq (0,0) \qquad (1)$$

and a 2-D data sequence with $K \times L$ points as shown in Fig. 1(a). We assume that the entire data set is

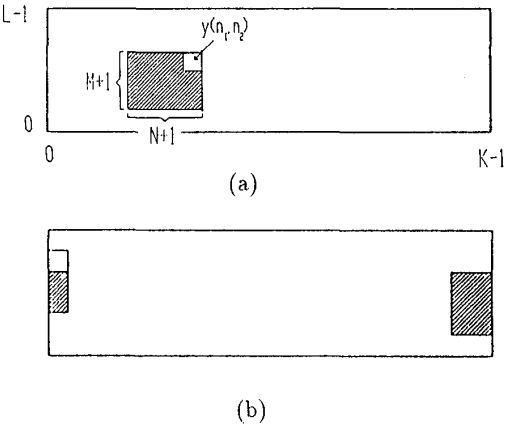

Figure 1: First Quadrant $(N+1) \times (M+1)$ Filter

processed by scanning along rows and that in processing the data, the data is considered to 'wrap around' at the end points in a helicoidal fashion, so the prediction filter has the support depicted in Fig. 1(b) at the beginning and ends of rows. This is an important assumption in developing the filter update procedure. If the prediction error is defined as

$$e(n_1,n_2) = y(n_1,n_2) - \hat{y}(n_1,n_2) \qquad (2)$$

then the optimal least-squares filter is defined to be the filter that minimizes the accumulated squared error

$$\epsilon(n_1, n_2) = \sum_{i=0}^{n_1} [e(i, n_2)]^2 + \sum_{i=0}^{K-1} \sum_{j=0}^{n_2-1} [e(i,j)]^2 \qquad (3)$$

The purpose of this paper is to describe a fast RLS algorithm that solves this problem.

Because of limited space and the length and complexity of the derivation we will not be able to *derive* the algorithm here. (A complete derivation is given in [4] and requires several pages.) The derivation relies heavily on the use of projection operations and geometrical concepts; these will be more difficult to appreciate in this limited discussion. In what follows we will merely define the components of the algorithm and give a brief outline of its steps.

2.2 Algorithm Components

The key to developing a fast 2-D RLS algorithm lies in the consideration of a forward and a backward multichannel filtering problem, and their interrelation with the 2-D problem. These components of the algorithm will be discussed with the aid of Fig. 2.

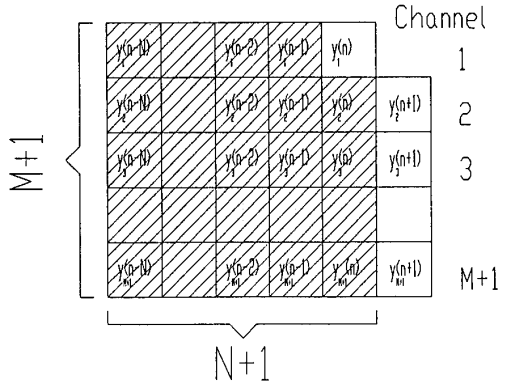

Figure 2: $M+1$ Channel Analogy

Let us consider a "channel" to be the i^{th} row covered by the mask and denote the data in the i^{th} channel by $y_i(n)$ where n is the linear scanning index $n = n_2K + n_1$. With this new notation, the 2-D linear prediction problem has the form

$$\hat{y}_1(n) = \underline{\mathbf{y}}_{1,N}^T(n)\,\underline{\mathbf{a}}(n) \qquad (4)$$

where $\underline{\mathbf{y}}_{1,N}(n)$ is the data covered by the filter mask (shaded in Fig. 2) and $\underline{\mathbf{a}}(n)$ is the correspondingly ordered vector of 2-D filter coefficients.

$$\underline{\mathbf{y}}_{1,N}(n) = [y_1(n-1), \cdots, y_1(n-N), \qquad (5)$$
$$y_2(n), \cdots, y_{M+1}(n), \cdots, y_{M+1}(n-N)]^T$$

$$\underline{\mathbf{a}}(n) = [a_{10}(n), \cdots, a_{N0}(n), a_{01}, \cdots, a_{N1}(n), \qquad (6)$$
$$\cdots, a_{0M}(n), \cdots, a_{NM}(n)]^T$$

and the prediction error (2) is given by

$$e_1(n) = y_1(n) - \hat{y}_1(n) \qquad (7)$$

Let $\mathbf{x}_F(n)$ be a (M+1)-channel signal formed by the data acquired by the 2-D mask when it is moved from n to $n+1$ (unshaded boxes in Fig. 2). The matrix $\underline{\mathbf{F}}(n)$ is the set of optimal filter coefficients for predicting this data from the data under the filter mask. This prediction is given by

$$\hat{\mathbf{x}}_F(n) = \mathbf{F}^T(n)\underline{\mathbf{y}}_{1,N}(n) \qquad (8)$$

and we define the corresponding prediction error covariance matrix as $\Sigma_F(n)$.

A corresponding backward multichannel problem can be defined as follows. Let $\mathbf{x}_B(n)$ represent the data *left out* by the 2-D mask when it is moved from time n to $n+1$ (data in left column in Fig. 2). Then proceeding as for the forward problem, define a matrix of prediction coefficients $\mathbf{B}(n)$ and the corresponding prediction error covariance matrix $\Sigma_B(n)$. This prediction is of the form

$$\hat{\mathbf{x}}_B(n) = \mathbf{B}(n)^T \underline{\mathbf{y}}_{0,N-1}(n) \qquad (9)$$

where $\underline{\mathbf{y}}_{0,N-1}(n)$ is the data covered by the mask in Fig. 2 when it is moved from time n to $n+1$. (This includes the unshaded boxes.)

As a final step we proceed along lines similar to what has been done for 1-D single channel problems in developing the fast RLS. We introduce the fixed $n+1$-dimensional vector

$$\underline{\pi}(n) = [0, 0, 0, \cdots, 0, 0, 0, 1]^T \qquad (10)$$

and consider the filter $\underline{\mathbf{g}}(n)$ necessary for predicting this vector using all of the data up to time n. The estimate of $\underline{\pi}(n)$ is of the form

$$\hat{\underline{\pi}}(n) = \mathbf{Y}_{0,N-1}(n)\underline{\mathbf{g}}(n) \qquad (11)$$

where $\mathbf{Y}_{0,N-1}(n)$ is the data matrix whose rows are the transposed vectors $\underline{\mathbf{y}}_{0,N-1}^T(k)$, $k = 0, \ldots, n$. This filter $\underline{\mathbf{g}}$ is known as the *gain transversal filter* and

its consideration leads to a method for quantifying the angular change between the subspaces associated with data matrices at times n and $n-1$. From geometric arguments it can be shown that

$$\gamma(n) = 1 - \underline{\mathbf{y}}_{0,N-1}^T(n)\underline{\mathbf{g}}(n) = \cos^2\theta \qquad (12)$$

where θ is the angle between $\underline{\pi}(n)$ and the vector normal to the data subspace. Updating this filter and the associated angular change parameter $\gamma(n)$ is key to updating the 2-D filter.

One further set of terms is necessary for the 2-D fast RLS problem. This is the concept of the *extended* gain transversal filter $\underline{\mathbf{g}}''(n)$. Its interpretation is that of estimating $\underline{\pi}(n)$ from the extended data sets $\{\underline{\mathbf{y}}_{0,N}(n)\}$ where $\underline{\mathbf{y}}_{0,N}(n)$ is *all* of the data shown in Fig. 2 at time n (both shaded and unshaded boxes). Since the data for $\underline{\mathbf{y}}_{0,N}(n)$ is defined by catenating rows of the extended area, consideration of the extended problem necessarily requires the use of forward and backward permutation matrices Ψ_F and Ψ_B which are defined implicitly by

$$\begin{aligned}\underline{\mathbf{y}}_{0,N}(n) &= [\mathbf{x}_F(n), \mathbf{y}_{1,N}(n)]\,\Psi_F \\ &= [\mathbf{y}_{0,N-1}(n), \mathbf{x}_B(n)]\,\Psi_B \end{aligned} \qquad (13)$$

Certain partitions $\mathbf{M}(n)$ and $\mathbf{m}(n)$ of the extended gain (see Eq. 20 below) corresponding to the partitioning $[\mathbf{y}_{0,N-1}(n), \mathbf{x}_B(n)]$ of the extended data are needed in the algorithm. Finally $\gamma'(n)$ is the angular parameter corresponding to the extended gain transversal vector.

2.3 Steps and Equations

To begin the algorithm the four filters $\underline{\mathbf{a}}$,$\mathbf{F}$, $\mathbf{B}$, and $\underline{\mathbf{g}}$ are set to zero for $n = 0$. The angle parameter $\gamma(0)$ is set to 1.0 since the initial data is zero and since all of the subspaces associated with previous data are the null space. Since a positive forward prediction error variance is necessary for the algorithm to start Σ_F^{-1} is set to a diagonal matrix with elements $1/\delta$ where δ is a small positive constant. (The backward prediction error covariance matrix Σ_B is not explicitly required in the recursion.) For the computational analysis we define $K_1 = M+1$ (the number of channels) and $K_2 = (M+1)(N+1) - 1$ (the number of 2-D filter parameters). The terms to be computed at each iteration, and their formulas are given below.

A priori 2-D prediction error (K_2 operations):

$$e_1(n|n-1) = y_1(n) - \underline{\mathbf{y}}_{1,N}^T(n)\underline{\mathbf{a}}(n-1) \qquad (14)$$

2-D filter update (K_2 operations):

$$\underline{\mathbf{a}}(n) = \underline{\mathbf{a}}(n-1) + \underline{\mathbf{g}}(n-1)e_1(n|n-1) \qquad (15)$$

A priori multichannel forward prediction error (K_1K_2 operations):

$$\mathbf{e}_F(n|n-1) = \mathbf{x}_F(n) - \mathbf{F}^T(n-1)\underline{\mathbf{y}}_{1,N}(n) \qquad (16)$$

Multichannel forward prediction error (K_1 operations):

$$\mathbf{e}_F(n) = \mathbf{e}_F(n|n-1)\gamma(n-1) \qquad (17)$$

Inverse error covariance matrix for the multichannel forward filter ($1.5K_1^2 + 2.5K_1$ operations):

$$\begin{aligned}\Sigma_F^{-1}(n) &= \Sigma_F^{-1}(n-1) \\ &\quad - \frac{\Sigma_F^{-1}(n-1)\mathbf{e}_F(n)\mathbf{e}_F^T(n)\Sigma_F^{-1}(n-1)}{\gamma(n-1) + \mathbf{e}_F^T(n)\Sigma_F^{-1}(n-1)\mathbf{e}_F(n)}\end{aligned} \qquad (18)$$

Multichannel forward filter update (K_1K_2 operations):

$$\mathbf{F}(n) = \mathbf{F}(n-1) + \underline{\mathbf{g}}(n-1)\mathbf{e}_F^T(n|n-1) \qquad (19)$$

Extended gain transversal filter ($K_1^2 + K_1K_2$ operations):

$$\begin{aligned}\underline{\mathbf{g}}''(n) &= \begin{bmatrix}\mathbf{M}(n)\\ \mathbf{m}(n)\end{bmatrix} \\ &= \Psi_B\Psi_F^T\left(\begin{bmatrix}\mathbf{0}_{M+1}\\ \underline{\mathbf{g}}(n-1)\end{bmatrix} + \begin{bmatrix}\mathbf{I}_{M+1}\\ -\mathbf{F}(n)\end{bmatrix}\Sigma_F^{-1}\mathbf{e}_F(n)\right)\end{aligned} \qquad (20)$$

(subscripts $M+1$ represent matrix dimensions)

Extended angle parameter (K_1 operations using previous results):

$$\gamma'(n) = \gamma(n-1) - \mathbf{e}_F^T(n)\Sigma_F^{-1}(n)\mathbf{e}_F(n) \qquad (21)$$

A priori multichannel backward prediction error (K_1K_2 operations):

$$\mathbf{e}_B(n|n-1) = \mathbf{x}_B^T(n) - \mathbf{B}^T(n-1)\underline{\mathbf{y}}_{0,N-1}^T(n) \qquad (22)$$

Angle parameter ($K_1 + 1$ operations):

$$\gamma(n) = \gamma'(n)[1 - \mathbf{e}_B^T(n|n-1)\mathbf{m}(n)]^{-1} \qquad (23)$$

Gain transversal filter ($K_1K_2 + K_2$ operations using previous results):

$$\begin{aligned}\underline{\mathbf{g}}(n) &= [\mathbf{M}(n) + \mathbf{B}(n-1)\mathbf{m}(n)] \\ &\quad \times\ (1 - \mathbf{e}_B^T(n|n-1)\mathbf{m}(n))^{-1}\end{aligned} \qquad (24)$$

Multichannel backward filter update (K_1K_2 operations):

$$\mathbf{B}(n) = \mathbf{B}(n-1) + \underline{\mathbf{g}}(n)\mathbf{e}_B^T(n|n-1) \qquad (25)$$

The total number of operations (multiplications or divisions) required per iteration by the algorithm is

$$2.5K_1^2 + 6K_1K_2 + 4.5K_1 + 3K_2 + 1 \qquad (26)$$

3 ARMA Model Extensions

The ARMA version of the 2-D fast RLS can be viewed as follows. Let us call the output or *observed* data $y_1(n)$ and the input data $w_1(n)$. For the present let us assume that this latter sequence is also known or observed. Let us separate the coefficients that operate on the two different sequences and call $\underline{\mathbf{a}}(n)$ the vector of AR coefficients of the filter, and $\underline{\mathbf{b}}(n)$ the vector of MA coefficients of the filter. As before, we develop this extension of the 2-D fast RLS to ARMA models assuming a first quadrant $(N+1)\times(M+1)$ quarter plane mask for both the AR and MA components of the filter, noting that more general forms are possible. Using the scanning index n defined before, we proceed by defining an ARMA prediction filter of the form

$$\hat{y}_1(n) = \underline{\mathbf{y}}_{1,N}^T(n)\,\underline{\mathbf{a}}(n) + \underline{\mathbf{w}}_{1,N}^T(n)\,\underline{\mathbf{b}}(n) \qquad (27)$$

with $\underline{\mathbf{y}}_{1,N}(n)$ and $\underline{\mathbf{w}}_{1,N}(n)$ defined using the same ordering as in (5). We want to find $\underline{\mathbf{a}}(n)$ and $\underline{\mathbf{b}}(n)$ to minimize the sum of squared errors

$$\epsilon_1(n) = \sum_{i=0}^{n}[e_1(i)]^2 \qquad (28)$$

where the prediction error is given by

$$e_1(n) = y_1(n) - \hat{y}_1(n) \qquad (29)$$

This can be written in vector notation as

$$\epsilon_1(n) = \underline{\mathbf{e}}_1^T(n)\underline{\mathbf{e}}_1(n) \qquad (30)$$

with

$$\underline{\mathbf{e}}_1(n) \;=\; \underline{\mathbf{y}}_1(n) - \hat{\underline{\mathbf{y}}}_1(n) \qquad (31)$$

We can combine the AR and MA coefficients in one single vector $\underline{\mathbf{c}}(n)$ as

$$\underline{\mathbf{c}}(n) \;=\; [\underline{\mathbf{a}}^T(n), \underline{\mathbf{b}}^T(n)]^T \qquad (32)$$

The data under the mask can then be expressed as

$$\underline{\mathbf{z}}_{1,N}(n) \;=\; [\underline{\mathbf{y}}_{1,N}^T(n), \underline{\mathbf{w}}_{1,N}^T(n)]^T \qquad (33)$$

Now we have for the estimate of $\underline{\mathbf{y}}_1(n)$

$$\hat{\underline{\mathbf{y}}}_1(n) = \mathbf{Z}_{1,N}(n)\,\underline{\mathbf{c}}(n) \qquad (34)$$

where $\mathbf{Z}_{1,N}(n)$ is the data matrix

$$\mathbf{Z}_{1,N}(n) = [\mathbf{Y}_{1,N}(n), \mathbf{W}_{1,N}(n)] \qquad (35)$$

formed by $\mathbf{Y}_{1,N}(n)$ and $\mathbf{W}_{1,N}(n)$, the data matrices whose rows are the transposed vectors vectors $\underline{\mathbf{y}}_{1,N}^T(k)$ and $\underline{\mathbf{w}}_{1,N}^T(k)$, $k=0,\ldots,n$. The least squares solution for $\underline{\mathbf{c}}(n)$ is given by the pseudo-inverse of $\mathbf{Z}_{1,N}(n)$

$$\underline{\mathbf{c}}(n) \;=\; \left(\mathbf{Z}_{1,N}^T(n)\mathbf{Z}_{1,N}(n)\right)^{-1}\mathbf{Z}_{1,N}^T(n)\underline{\mathbf{y}}_1(n) \qquad (36)$$

After defining new projection matrices and transversal filter operators associated with the new data matrices, the algorithm to recursively update $\underline{\mathbf{c}}(n)$ closely follows the procedures developed above.

4 Conclusions

A two-dimensional fast recursive least squares algorithm was described in this paper. The derivation is based on the relation between least squares prediction and the concepts of orthogonality associated with vector spaces. The ordering necessary to develop the recursive algorithm was imposed on the data by using a linear scanning index.

The algorithm has been compared in performance with a more standard form RLS algorithm for 2-D and a 2-D LMS algorithm [4] A substantial reduction in computational cost is obtained when compared with the basic 2-D RLS algorithm. The 2-D fast RLS algorithm requires on the order of $6K_1K_2$ arithmetic operations per iteration compared with $1.5K_2^2$ for the basic RLS, where K_1 is the number of channels defined for the 2-D fast RLS algorithm and K_2 is the total number of coefficients in the 2-D filter. The 2-D LMS algorithm due to its simplicity, is more economical than our algorithm in terms of computational cost, but lacks the excellent convergence performance experienced for the 2-D fast RLS.

References

[1] S. T. Alexander. Fast adaptive filters: a geometrical approach. *IEEE ASSP Magazine*, October 1986.

[2] Benjamin Friedlander, Daniel T. L. Lee, and Martin Morf. Recursive least squares ladder estimation algorithms. *IEEE Transactions on ASSP*, ASSP-29:627–641, June 1981.

[3] Thomas Kailath, Hanoch Lev-Ari, and John Cioffi. Least squares adaptive lattice and transversal filters: a unified geometric theory. *IEEE Transactions on Information Theory*, IT-30:222–236, March 1984.

[4] Armando M. Sequeira. *Adaptive Two Dimensional RLS Algorithms.* E.E. thesis, Naval Postgraduate School, March 1989.