A UNIFIED DERIVATION OF SQUARE-ROOT MULTICHANNEL LEAST-SQUARES FILTERING ALGORITHMS*

BABAK H. KHALAJ , ALI H. SAYED AND THOMAS KAILATH

Information Systems Laboratory, Stanford University, Stanford, CA 94305, USA.

ABSTRACT

We present a new derivation of exact least-squares multichannel and multi-dimensional adaptive algorithms, based on explicitly formulating the problem as a state-space estimation problem and then using different square-root versions of the Kalman, Chandrasekhar, and information algorithms. Moreover, by exploiting the shift structure of the input data, we further derive the fast multichannel RLS and lattice filters within the same framework.

1. INTRODUCTION

Multichannel least-squares algorithms are common in many areas of signal processing, such as two-dimensional signal processing and image-enhancement [1, 2], nonlinear filtering [3], and adaptive beamforming. The associated exact least-squares recursive methods basically fall into 3 main classes: transversal filters and their corresponding fast versions [1, 4], least squares lattice (LSL) filters [2, 5], and QR-based algorithms [6, 7]. The algorithms are usually derived via a variety of methods such as matrix partitioning and matrix inversion lemma, geometric projection theory, simultaneous solution of forward and backward prediction problems, etc.. The derivations, even in the single-channel case, are quite lengthy and their relationships are normally obscured by the different approaches.

Sayed and Kailath have recently proposed [8, 9, 10] a unified square-root based derivation of (single channel) adaptive filtering schemes that is based on reformulating the original problem as a state-space linear least-squares estimation problem. In this process, rich connections are encountered with algorithms that have been long established in linear estimation theory, such as the Kalman filtering algorithm, the square-root versions of the Kalman equations, the Chandrasekhar recursions and their variations, and the information forms of the Kalman and Chandrasekhar algorithms. In this paper we extend the aforementioned derivation to the multichannel adaptive filtering case, which we also reformulate as a state-space estimation problem. The amount of data to be processed here is usually significantly higher than in the single-channel case and hence, reducing the computational complexity of the standard multichannel RLS algorithm is of major importance. This reduction is usually achieved by invoking the existing shift structure in the input data. For this purpose, we shall show how to apply the so-called extended Chandrasekhar recursions [10, 11], with an appropriate choice of the initial covariance matrix, in order to reduce the computations by an order of magnitude.

In multichannel filters, the number of weights in different channels is not necessarily the same. We shall illustrate this with 2 examples: a nonlinear Volterra-series filter and a two-dimensional filter; in the former case the number of weights varies among the channels, but in the latter case all channels have the same number of weights.

2. MULTICHANNEL LEAST-SQUARES FILTERING

A general multichannel filter consists of M channels, where the desired response $d(i)$, assumed scalar for convenience of exposition, is estimated by a linear combination of the outputs of these channels. If $\mathbf{u}_i^{(k)}$ is the input vector to the kth channel at time i, then the linear least-squares estimate of the desired response $d(i)$ is given by $\hat{d}(i) = \sum_{k=1}^{M} \mathbf{u}_i^{(k)} \mathbf{w}^{(k)}$, where $\mathbf{w}^{(k)} = [w_1^{(k)} \ \ldots \ w_{N_k}^{(k)}]^T$ is the weight vector of the kth channel. In general, the number of weights (N_k) can vary among the channels and we shall denote the total number of weights of the filter by $N_t = \sum_{k=1}^{M} N_k$ By arranging all inputs and weights in $N_t \times 1$ column vectors $\mathbf{u}_i^T$ and $\mathbf{w}$, the multichannel problem can be reduced to a single-channel RLS problem, where it is desired to minimize the sum $\sum_{i=0}^{N} \lambda^{N-i} |d(i) - \mathbf{u}_i \mathbf{w}|^2$, where $N+1$ is the total number of data. By a rescaling of variables, viz., by defining $y(i) = d(i)/(\sqrt{\lambda})^i$ and $\mathbf{x}_i = \mathbf{w}/(\sqrt{\lambda})^i$, the above minimization problem can be recast into a Kalman filtering framework by introducing an N_t-dimensional state-space model of the form (see [8, 9] for a related discussion in the single channel case):

* This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command under Contract AFOSR91-0060, and by the SDIO/IST Program managed by the Army Research Office under Contract DAAL03-90-G-0108. The work of the second author was also supported by a fellowship from Fundação de Amparo à Pesquisa do Estado de São Paulo and by Escola Politécnica da Universidade de São Paulo, Brazil.

$$\begin{aligned} \mathbf{x}_{i+1} &= \lambda^{-1/2}\mathbf{x}_i, \quad \mathbf{x}_0 = \mathbf{w}, \quad \Pi_0 = \infty I \\ y(i) &= \mathbf{u}_i\mathbf{x}_i + v(i), \quad Ev(i)v^*(j) = \delta_{ij} \end{aligned} \tag{1}$$

Applying the Kalman filtering algorithm to the above state-space model gives an estimate of the state vector at time i. This estimate is related to the estimate of the weight vector via $\mathbf{w}_i = (\sqrt{\lambda})^{i+1}\hat{\mathbf{x}}_{i+1|i}$. In fact a direct correspondence between the Kalman and RLS variables can be established in this way [8], and the Kalman filter equations lead to the usual RLS algorithm (see, *e.g.*, [12, page 483]).

3. FAST MULTICHANNEL RLS

In several applications, the input channels $\mathbf{u}_i^{(k)}$ exhibit a shift structure of the form

$$\mathbf{u}_i^{(k)} = \left[u^{(k)}(i) \quad u^{(k)}(i-1) \quad \ldots \quad u^{(k)}(i-N_k+1)\right]$$

In this case the multichannel filter is constructed from M transversal filters where the linear least-squares estimate of $d(i)$ is given by

$$\hat{d}(i) = \sum_{k=1}^{M}\sum_{j=1}^{N_k} w_j^{(k)}(i)u^{(k)}(i-j+1)$$

To exploit the shift structure of the multichannel filter, we form a state-space model of dimension $M(N+1)$ as follows:

$$\mathbf{x}_{i+1} = \lambda^{-1/2}\mathbf{x}_i, \quad y(i) = \mathbf{h}_i\mathbf{x}_i + v(i) \tag{2}$$

$$\mathbf{x}_0 = [\; \mathbf{w}^{(1)T} \quad 0_{N-N_1+1} \quad \ldots \quad \mathbf{w}^{(M)T} \quad 0_{N-N_M+1} \;]^T$$

$$\mathbf{h}_i = [\; \underbrace{u^{(1)}(i) \;\ldots\; u^{(1)}(0) \;\; 0_{N-i}}_{1st\ Channel} \;\ldots\; \underbrace{u^{(M)}(i) \;\ldots\; u^{(M)}(0) \;\; 0_{N-i}}_{Mth\ Channel} \;]$$

The Kalman filter of the above state-space model is given by

$$\hat{\mathbf{x}}_{i+1|i} = \lambda^{-1/2}\hat{\mathbf{x}}_{i|i-1} + \mathbf{k}_i r_{e,i}^{-1}\left[y(i) - \mathbf{h}_i\hat{\mathbf{x}}_{i|i-1}\right]$$

$$r_{e,i} = 1 + \mathbf{h}_i P_{i|i-1}\mathbf{h}_i^*, \quad \mathbf{k}_i = \lambda^{-1/2}P_{i|i-1}\mathbf{h}_i^*$$

$$P_{i+1|i} = \lambda^{-1}\left[P_{i|i-1} - P_{i|i-1}\mathbf{h}_i^* r_{e,i}^{-1}\mathbf{h}_i P_{i|i-1}\right] \tag{3}$$

The initial state-covariance matrix in the multichannel case will be a block matrix of the form,

$$\begin{aligned} P_{0/-1} &= E[\mathbf{x}_0 - \hat{\mathbf{x}}_{0/-1}][\mathbf{x}_0 - \hat{\mathbf{x}}_{0/-1}]^* = \\ &= \begin{bmatrix} \pi_0^{(11)} & 0 & \ldots & \pi_0^{(1M)} & 0 \\ 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ \pi_0^{(M1)} & 0 & \ldots & \pi_0^{(MM)} & 0 \\ 0 & 0 & \ldots & 0 & 0 \end{bmatrix} \end{aligned} \tag{4}$$

where each $\pi_0^{(kl)}$ is an $N_k \times N_l$ matrix, and the diagonal matrices $\pi_0^{(kk)}$ should be positive definite.

Note that the block vector $\mathbf{h}_i$ satisfies the relation, $\mathbf{h}_i = \mathbf{h}_{i+1}\Psi$, where $\Psi = Z \oplus Z \oplus \ldots \oplus Z$ is a block shift matrix. Due to the shift structure of $\mathbf{h}_i$, viz., $\mathbf{h}_i = \mathbf{h}_{i+1}\Psi$, the proposed state-space model (2) falls into the class of ***structured time-variant*** models for which the extended Chandrasekhar recursions [9, 11] can be used to reduce the order of computations. These recursions are based on the factorization $P_{i+1|i} - \Psi_i P_{i|i-1}\Psi_i^* \equiv L_i S L_i^*$, where L_i is a low rank $M(N+1)\times\alpha$ matrix. Due to the special block structure of $P_{0|-1}$ and $\mathbf{h}_i$, the matrix L_0 and the gain vector $\bar{\mathbf{k}}_{p,i} = \mathbf{k}_i r_{e,i}^{-*/2}$, will be of the form:

$$L_0 = \left[\; \tilde{L}_0^{(1)T} \quad 0 \quad \ldots \quad \tilde{L}_0^{(M)T} \quad 0 \;\right]^T,$$

$$\bar{\mathbf{k}}_{p,i} = [\; \mathbf{c}_i^{(1)T} \quad 0_{N-N_1+1} \quad \ldots \quad \mathbf{c}_i^{(M)T} \quad 0_{N-N_M+1} \;]^T$$

where $\tilde{L}_0^{(j)}$ is $(N_j+1)\times\alpha$.

By defining $\tilde{\mathbf{h}}_i^{(j)}$ to be a row vector of the first N_j+1 coefficients of the jth channel of $\mathbf{h}_i$, the extended Chandrasekhar recursions for the structured state-space model (2) then leads to the following fast RLS algorithm in square-root array form:

$$\begin{bmatrix} r_{e,i}^{1/2} & \sum_{j=1}^{M}\tilde{\mathbf{h}}_{i+1}^{(j)}\tilde{L}_i^{(j)} \\ \begin{bmatrix} 0 \\ \mathbf{c}_i^{(1)} \\ \vdots \\ 0 \\ \mathbf{c}_i^{(M)} \end{bmatrix} & \lambda^{-1/2}\begin{bmatrix} \tilde{L}_i^{(1)} \\ \vdots \\ \tilde{L}_i^{(M)} \end{bmatrix} \end{bmatrix}\Theta_i = \begin{bmatrix} r_{e,i+1}^{1/2} & 0_{1\times\alpha} \\ \begin{bmatrix} \mathbf{c}_{i+1}^{(1)} \\ 0 \\ \vdots \\ \mathbf{c}_{i+1}^{(M)} \\ 0 \end{bmatrix} & \begin{bmatrix} \tilde{L}_{i+1}^{(1)} \\ \vdots \\ \tilde{L}_{i+1}^{(M)} \end{bmatrix} \end{bmatrix} \tag{5}$$

where Θ_i is any $(1\oplus S)$-unitary matrix that produces the block zero at the right hand side of the above equation. To apply these recursions, we first have to compute a (nonunique) factor L_0 defined by

$$L_0 S L_0^* \;=\; \lambda^{-1}\left(P_{0|-1} - \bar{\mathbf{k}}_{p,0}\bar{\mathbf{k}}_{p,0}^*\right) - \Psi P_{0|-1}\Psi^* \tag{6}$$

where S is an $\alpha\times\alpha$ signature matrix. In this way, the order of computations reduces to $O(\alpha(N_t+M))$. The value of α (rank of L_0) depends on the choice of the a priori covariance matrix $\Pi_0 = P_{0|-1}$. The choice $\Pi_0 \to \infty$ guarantees that the limiting case of the stochastic Kalman filter is the deterministic RLS problem. But in practice other choices of Π_0 are of interest and they affect the complexity of the algorithm (see also [13, 14]). For example, there are cases where it is preferable to start the recursive algorithm with a nonzero initial value $\mathbf{x}_0$; in order to change the tracking capability of the algorithm or to start with an initial guess obtained by another method.

By choosing a finite Π_0 the total error criterion is then given by:

$$\min_{\mathbf{w}} \sum_{i=0}^{N} \lambda^{N-i}|d(i) - \mathbf{u}_i\mathbf{w}|^2 + (\mathbf{w}-\bar{\mathbf{w}})^*\Pi_0^{-1}(\mathbf{w}-\bar{\mathbf{w}})$$

with, $\bar{\mathbf{w}} = E\mathbf{w}$, $E(\mathbf{w}-\bar{\mathbf{w}})(\mathbf{w}-\bar{\mathbf{w}})^* = \Pi_0$. The solutions is still the filter equations (5), but with initial conditions $P_{0|-1} = \Pi_0$ and $\hat{\mathbf{x}}_{0|-1} = \bar{\mathbf{w}}$.

3.1 2D FILTERING

In 2D filtering problems, a large class of real-world images can be modeled as [15] a 2D auto-regressive process of the form:

$$u(i,j) = \sum_{(k,l)\in R} w_{kl}u(i-k, j-l) + n(i,j)$$

By arranging the elements of R in a block vector $\mathbf{u}_i$, it can be shown that $\mathbf{u}_i$ has a shift structure of the form $\mathbf{u}_i = \mathbf{u}_{i+1}\Psi$, where Ψ is a block shift matrix. For example, Fig. 1 shows the case of a casual window, where each row of the window is considered as a separate channel. At each step, all channels are shifted separately by one pixel, and by putting the rows in a block vector, the aforementioned shift structure is obtained.

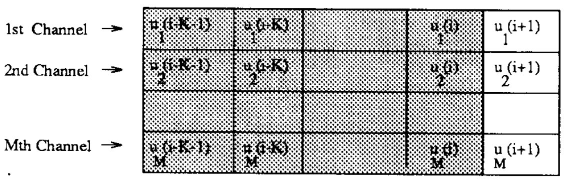

Figure 1: The 2D casual model and the corresponding channels

In this case all channels have the same number of weights ($N_i = K$), and all $\Pi_0^{(kl)}$ matrices will be $K \times K$ square matrices. The prewindowed case, and assuming no correlation between different channels, leads to a diagonal choice for $P_{0|-1}$. In order to obtain a low value for α, we choose $\Pi_0^{(kl)}$ as follows:

$$\Pi_0^{(kl)} = \begin{cases} diag\{\lambda, \lambda^2, \ldots, \lambda^K\} & k = l \\ 0 & k \neq l \end{cases}$$

then $(\lambda^{-1}P_{0|-1} - \Psi P_{0|-1}\Psi^*)$ will have rank $2M$ ($\alpha = 2M$) and will be equal to

$$\begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix}$$

where A is the block-diagonal matrix

$$A = \begin{bmatrix} \begin{pmatrix} 1 & & \\ & 0 & \\ & & -\lambda^K \end{pmatrix} & 0 \\ 0 & 0 \end{bmatrix}$$

So the above choice of the prior covariance matrix results in an algorithm of complexity $O(2MN_t)$, which is the complexity of the existing fast multichannel transversal filters [4, 7]. But the order of computations can be further reduced by other selections of the prior covariance matrix. In fact, by choosing $\Pi_0^{(kl)} = diag\{\lambda, \lambda^2, \ldots, \lambda^K\}$ for all k, l, the expression $\lambda^{-1}P_{0|-1} - \Psi P_{0|-1}\Psi^*$ will be equal to:

$$\begin{bmatrix} A & A & \ldots & A \\ A & A & \ldots & A \\ \vdots & \vdots & \ddots & \vdots \\ A & A & \ldots & A \end{bmatrix},$$

which leads to a rank 2 matrix ($\alpha = 2$). In this way we obtain a fast 2D filter with computational complexity $O(2N_t)$, where N_t is equal to the size of the window or MK. For example, for a 2D filter with a square window ($M = K$), this special choice of $P_{0|-1}$ reduces the order of computations from $O(M^3)$ to $O(M^2)$. Therefore, this choice of Π_0 exploits the input shift structure better than the existing algorithms and reduces the computational load by an order of magnitude.

3.2 NONLINEAR FILTERING

Another example arises in the problem of adaptive nonlinear Volterra-series filtering [16], where the input-output relationship of a $2nd$ order filter of degree N is given by:

$$y(n) = \sum_{i=0}^{N} w_i x(n-i) + \sum_{i=0}^{N}\sum_{j=0}^{N} w_{ij}x(n-i)x(n-j)$$

By proper arrangement of the inputs into M channels (where $M = N+2$), the shift structure of the filter can be fully exploited (Fig.2).

1st Channel	→	$x(n)$	...	$x(n-N)$
2nd Channel	→	$x^2(n)$	...	$x^2(n-N)$
3rd Channel	→		...	$x(n-N+1)x(n-N)$
⋮			⋱	⋮
Mth Channel	→			$x(n)x(n-N)$

Figure 2: Shift structure of a Volterra filter

In this case the number of weights in all the channels (except the first two) is different, $N_1 = N_2 = N+1$, $N_i = N - i + 3$, $(3 \leq i \leq M)$. By choosing $\Pi_0^{(kk)} = diag\{\lambda, \lambda^2, \ldots, \lambda^{N+2-k}\}$ and $\Pi_0^{kl} = \mathbf{0}$ for $k \neq l$, α will be $2M$. Since the total number of weights for the nonlinear filter is $O(N^2)$, the overall order of computations of the fast nonlinear filter will be $O(N^3)$, which is the same as the complexity of the existing fast nonlinear adaptive filters.

4. MULTICHANNEL QR AND LATTICE

The multichannel lattice algorithms can also be derived from the original state-space model by applying the square-root *information* form of the Kalman filter [8, 17], which directly leads to the QR algorithm

$$\begin{bmatrix} \sqrt{\lambda}\Phi_{i-1}^{1/2} & \mathbf{u}_i^* \\ \sqrt{\lambda}\alpha_{i-1}^* & d^*(i) \\ 0 & 1 \end{bmatrix} \Theta_i = \begin{bmatrix} \Phi_i^{1/2} & 0 \\ \alpha_i^* & e^*(i)\gamma_i^{1/2} \\ \mathbf{u}_i\Phi_i^{-*/2} & \gamma_i^{1/2} \end{bmatrix} \quad (7)$$

where $e(i)$ is the a-priori error, γ_i is the conversion factor, $\alpha_i = \Phi_i^{-1/2}\theta_i$, $\Phi_i = \sum_{j=0}^{i}\lambda^{i-j}\mathbf{u}_j^*\mathbf{u}_j$, $\theta_i = \sum_{j=0}^{i}\lambda^{i-j}d(j)\mathbf{u}_j^*$. Hence, the least-squares error due to estimating $d(i)$ can be obtained via an array equation as above. This observation readily allows us to derive the corresponding lattice array equations as follows: consider the case of a multichannel transversal filter where all the channels have the same number of weights. Then *vector* forward and backward prediction errors can be defined as [5],

$$\mathbf{f}_i^k = \mathbf{u}(i) - \hat{\mathbf{u}}(i|i-1:i-k)$$

$$\mathbf{b}_{i-1}^k = \mathbf{u}(i-k-1) - \hat{\mathbf{u}}(i-k-1|i-1:i-k)$$

where $\mathbf{u}(i) = [u^{(1)}(i)\ u^{(2)}(i)\ \ldots\ u^{(M)}(i)]^T$, consists of the most recent inputs of all channels. Knowing $\mathbf{f}_i^k$ we shall try to determine $\mathbf{f}_i^{k+1}$, which is the forward prediction error at time i in estimating $\mathbf{u}(i)$ from $\mathbf{u}(i-k-1), \ldots, \mathbf{u}(i-1)$. Notice that we are using the same input data as $\mathbf{f}_i^k$, except for $\mathbf{u}(i-k-1)$, and so we only need to incorporate the new information that is in $\mathbf{u}(i-k-1)$ and not in the previous data. But this new information is nothing more than $\mathbf{b}_{i-1}^k$. So, $\mathbf{f}_i^{k+1}$ is the least-squares error of estimating $\mathbf{f}_i^k$ using $\mathbf{b}_{i-1}^k$, and can be propagated using an array structure similar to (7) :

$$\begin{bmatrix} \sqrt{\lambda}\bar{\Phi}_{k,i-2}^{b/2} & \bar{\mathbf{b}}_{i-1}^{*k} \\ \sqrt{\lambda}\bar{\alpha}_{k,i-2}^{*b} & \bar{\mathbf{f}}_i^{*k} \end{bmatrix} \bar{\Theta}_{k,i-1}^b = \begin{bmatrix} \bar{\Phi}_{k,i-1}^{b/2} & 0 \\ \bar{\alpha}_{k,i-1}^{*b} & \bar{\mathbf{f}}_i^{*(k+1)} \end{bmatrix}$$

where $\bar{\mathbf{f}}_i^{k+1}$ and $\bar{\mathbf{b}}_{i-1}^k$ are normalized forward and backward prediction error vectors of order k at times i and $i-1$ respectively, $\bar{\alpha}_{k,i}^b = \bar{\Phi}_{k,i}^{-b/2}\bar{\theta}_{k,i}^b$, $\bar{\Phi}_{k,i}^b = \sum_{j=0}^{i} \lambda^{i-j}\bar{\mathbf{b}}_j^k\bar{\mathbf{b}}_j^{*k}$, $\bar{\theta}_{k,i}^b = \sum_{j=0}^{i} \lambda^{i-j}\bar{\mathbf{f}}_{j+1}^k\bar{\mathbf{b}}_j^{*k}$.

Backward errors can be obtained by following a similar argument leading to

$$\begin{bmatrix} \sqrt{\lambda}\Phi_{k,i-1}^{f/2} & \bar{\mathbf{f}}_i^{*k} \\ \sqrt{\lambda}\bar{\alpha}_{k,i-1}^{*f} & \bar{\mathbf{b}}_{i-1}^{*k} \end{bmatrix} \bar{\Theta}_{k,i}^f = \begin{bmatrix} \bar{\Phi}_{k,i}^{f/2} & 0 \\ \bar{\alpha}_{k,i}^{*f} & \bar{\mathbf{b}}_i^{*(k+1)} \end{bmatrix}$$

where the normalized quantities are defined as $\bar{\Phi}_{k,i}^f = \sum_{j=0}^{i} \lambda^{i-j}\bar{\mathbf{f}}_j^k\bar{\mathbf{f}}_j^{*k}$, $\bar{\theta}_{k,i}^f = \sum_{j=0}^{i} \lambda^{i-j}\bar{\mathbf{b}}_{j-1}^k\bar{\mathbf{f}}_j^{*k}$, $\bar{\alpha}_{k,i}^f = \bar{\Phi}_{k,i}^{-f/2}\bar{\theta}_{k,i}^f$

The joint process estimation can be performed by using the sequence of orthogonal backward prediction errors $\{\bar{\mathbf{b}}_i^k\}$ and the array structure:

$$\begin{bmatrix} \sqrt{\lambda}\bar{\Phi}_{k,i-2}^{b/2} & \bar{\mathbf{b}}_{i-1}^{*k} \\ \sqrt{\lambda}\bar{\alpha}_{k,i-2}^{*d} & \bar{e}_{d,i-1}^{*k} \end{bmatrix} \bar{\Theta}_{k,i-1}^b = \begin{bmatrix} \bar{\Phi}_{k,i-1}^{b/2} & 0 \\ \bar{\alpha}_{k,i-1}^{*d} & \bar{e}_{d,i-1}^{*(k+1)} \end{bmatrix}$$

where $\bar{\alpha}_{k,i}^d = \bar{\Phi}_{k,i}^{-b/2}\bar{\theta}_{k,i}^d$, and $\bar{\theta}_{k,i}^d = \sum_{j=0}^{i} \lambda^{i-j}\bar{\mathbf{b}}_j^{*k}\bar{e}_{d,j}^k$. In this way, $d(i)$ is estimated by a linear combination of the backward prediction errors.

In the cases where the number of weights of the different channels is not the same, the backward and forward prediction errors of different sections of the lattice filter are of different dimensions, and a simple extension of the scalar case will not work any more. This problem can be solved by an appropriate definition of backward and forward prediction errors and by appending a set of auxiliary terms to the prediction error vectors at each stage [5]. It can be shown [18] that the computation of these auxiliary terms has a form similar to the joint process estimation problem and can be done using the basic array structure proposed here.

5. CONCLUDING REMARKS

We presented a unified derivation of multichannel adaptive filtering algorithms by proper identification with well-known algorithms in state-space linear least-squares estimation theory. We further remark that the generalized shift structure allowed by the extended Chandrasekhar recursions allows us to consider more general structures in the input data [8, 18].

REFERENCES

[1] Y. Boutalis, S. Kollias, and G. Carayannis, A fast multichannel approach to adaptive image estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37(7)**:1090–1098, July 1989.

[2] X. Liu, P. Baylou, and M. Najim, A new 2D fast lattice RLS algorithm, In *Proc. of IEEE ICASSP*, pages (III)329–332, 1992.

[3] Mushtaq A. Syed and V. John Mathews, QR-decomposition based algorithms for adaptive Volterra filtering, In *Proc. of IEEE ISCAS*, pages 2625–2628, 1992.

[4] D. Slock, L. Chisci, H. Lev-Ari, and T. Kailath, Modular and numerically stable fast transversal filters for multichannel and multiexperiment RLS, *IEEE Transactions on Signal Processing*, **40(4)**:784–802, April 1992.

[5] F. Ling and J. Proakis, A generalized multichannel least-squares lattice algorithm based on sequential processing, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **32(2)**:381–389, August 1984.

[6] P. Lewis, QR-based algorithms for multichannel adaptive least-squares lattice filters, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **38(3)**:421–432, March 1990.

[7] B. Yang and J. Bohme, Rotation-based RLS algorithms: Unified derivations, numerical properties, and parallel implementations, *IEEE Transactions on Signal Processing*, **40(5)**:1151–1167, May 1992.

[8] A. H. Sayed and T. Kailath, "A state-space approach to adaptive filtering", In *Proc. of this conference.*

[9] A. H. Sayed, *Displacement Structure in Signal Processing and Mathematics*, PhD thesis, Stanford University, Stanford, CA, August 1992.

[10] A.H. Sayed and T.Kailath, "Structured matrices and fast RLS adaptive filtering", In *2nd IFAC Workshop on Algorithms and Architectures for Real-Time Control*, Seoul,Korea, September 1992.

[11] A. H. Sayed and T. Kailath, "Extended Chandrasekhar recursions", *IEEE Transactions on Automatic Control*, to appear.

[12] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, second edition, 1991.

[13] A. Houacine, Regularized fast recursive least squares algorithms for adaptive filtering, *IEEE Transactions on Signal Processing*, **39**(4):860–870, April 1991.

[14] D. T. M. Slock, *Fast Algorithms for Fixed-Order Recursive Least-Squares Parameter Estimation*, PhD thesis, Stanford University, Stanford, CA, 1989.

[15] A. Jain, Advances in mathematical models for image processing, *Proceedings IEEE*, **37(7)**:1090–1098, July 1981.

[16] M. Schetzen, *The Volterra and Wiener Theory of Nonlinear Systems*, John Wiley, New York, 1980.

[17] M. Morf and T. Kailath, "Square root algorithms for least squares estimation", *IEEE Transactions on Automatic Control*, **20(4)**:487–497, August 1975.

[18] B. H. Khalaj, A. H. Sayed, and T. Kailath, "A unified derivation of square-root multichannel least-squares filtering algorithms", to be submitted.