

Order-Recursive RLS Laguerre Adaptive Filtering

Ricardo Merched, *Student Member, IEEE*, and Ali H. Sayed, *Senior Member, IEEE*

Abstract—This paper solves the problem of designing recursive-least-squares (RLS) lattice (or order-recursive) algorithms for adaptive filters that do not involve tapped-delay-line structures. In particular, an RLS-Laguerre lattice filter is obtained.

Index Terms—Laguerre network, lattice filter, order-recursive filter, regularized least-squares, RLS algorithm.

I. INTRODUCTION

THIS PAPER solves the problem of designing recursive-least-squares (RLS) lattice algorithms for adaptive filters that do not involve tapped-delay-line structures. As is well-known, all the derivations that are available so far in the literature for RLS order-recursive filters are based on the assumption of regression vectors with shift structure (see, e.g., [1]–[6]). The resulting filters are therefore not applicable to situations that involve other filter structures, such as Laguerre-based networks, where successive regression vectors are not shifted versions of each other.

In recent works [7], [8], it was shown that fast *fixed-order* RLS algorithms can be derived for certain more general structures in the regression vectors, other than the shift structure. These extended fast array methods turn out to be generalizations of earlier well-known fast transversal schemes for tapped-delay lines known as the fast *a posteriori* error sequential technique (FAEST) [9] and the fast transversal filter algorithm (FTF) [10]. As an example of the usefulness of these extensions, it was recently shown in [11] that the regression vectors that arise in a Laguerre-based network satisfy the structural conditions of [7] and that, therefore, an efficient fixed-order RLS scheme for updating the coefficients of a Laguerre-based adaptive filter can indeed be derived.

These results motivate us to pursue here the development of *order-recursive*, as opposed to fixed-order, adaptive algorithms for certain general filter structures, other than the conventional FIR structure. A consequence of our derivation will be the first RLS Laguerre-based lattice filter. While the existing RLS-based Laguerre solutions are all $\mathcal{O}(M^2)$ algorithms (e.g., [12]), with M being the order of the filter, the lattice filter of this paper offers an $\mathcal{O}(M)$ solution for the exact same problem. This result is useful especially since it has been realized for some time that Laguerre networks offer superior modeling capabilities when

compared with FIR networks at a reduced number of tap coefficients and with a guaranteed stable performance. This is in contrast with some adaptive IIR filter implementations that require stability monitoring. Excellent accounts of adaptive IIR filters and of the convenience of Laguerre-based filters can be found in [13]–[18]. An example of an application in echo cancellation appears in [19]. In particular, assuming stationary data, [18] proposes an LMS-like Laguerre-based lattice filter that is a generalization of the so-called gradient adaptive lattice (GAL) algorithm (see [5]).

In this paper, we derive an exact RLS Laguerre-based lattice algorithm. One advantage of the RLS-based algorithm, besides optimality, is that least-squares methods offer considerably superior convergence performance and lower misadjustment when compared with stochastic gradient solutions (see the simulation results in Section V and, in particular, Fig. 6).

We start our discussions in Section II with a brief review of the regularized least-squares problem, followed by derivations in Sections III of several order- and time-update relations. Although most of the expressions in this section may look familiar to readers acquainted with the theory of least-squares lattice filters, our presentation actually has three contributions. First, all expressions are derived *without* assuming any underlying structure in the regression vectors. The derivation of some of the relations derived in this section has been restricted in the literature to the case of shift structured data. Second, the derivation shows that it is possible to derive *efficient* order-recursive RLS filters, even for cases where the regression vectors do not possess shift structure. This is achieved by pointing out the exact variable whose update is intimately affected by the data structure. The derivation also shows what kinds of data structure lead to fast order-recursive filters. Finally, all order-recursive relations are derived by explicitly solving *regularized* least-squares problems from the start. In contrast, similar relations have always been derived in the literature without taking into account the need for regularization; this need is usually accounted for by initializing the lattice algorithm with certain nonzero initial conditions. Our arguments will show that these two ways of handling the initialization issue lead to different interpretations of some of the variables in the resulting algorithms.

We end our discussions with a derivation of the RLS Laguerre-lattice filter in Section IV. The algorithm is summarized in Table I, and its schematic representation is shown in Fig. 4. Comparing with the classical lattice filter of Fig. 2, we see that the new RLS Laguerre lattice filter still has a similar cascade structure. The main difference is that the delay blocks of Fig. 2 are replaced by a parallel lattice filter. This essentially amounts to replacing each delay element by a simple time-variant lattice section. We provide simulation results in Section V.

Manuscript received November 30, 1999; revised May 15, 2000. This work was supported in part by the National Science Foundation under Awards CCR-9732376 and ECS-9820765 and by UC CoRe under project CR 98-lock19. The work of R. Merched was also supported by a fellowship from CAPES, Brazil. The associate editor coordinating the review of this paper and approving it for publication was Prof. José Carlos M. Bermudez.

The authors are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: sayed@ee.ucla.edu).

Publisher Item Identifier S 1053-587X(00)07685-6.

II. REGULARIZED LEAST-SQUARES PROBLEM

We first provide a brief review of the regularized least-squares problem. Thus, given a column vector $y \in \mathbb{C}^{N+1}$ and a data matrix $H \in \mathbb{C}^{(N+1) \times M}$, the exponentially-weighted least squares problem seeks the column vector $w \in \mathbb{C}^M$ that solves

$$\min_w [\mu \lambda^{N+1} \|w\|^2 + (y - Hw)^* W (y - Hw)] \quad (1)$$

where μ is a scalar positive regularization parameter (usually small), and $W = (\lambda^N \oplus \lambda^{N-1} \oplus \dots \oplus 1)$ is a weighting matrix that is defined in terms of a forgetting factor λ satisfying $0 \ll \lambda < 1$. The symbol $*$ denotes complex conjugate transposition.

The individual entries of y will be denoted by $\{d(i)\}$, and the individual rows of H will be denoted by $\{u_i\}$

$$y = \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(N) \end{bmatrix}, \quad H = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix}.$$

Let w_N denote the optimal solution of (1). It is given by

$$w_N = (\mu \lambda^{N+1} I + H^* W H)^{-1} H^* W y \\ \triangleq P_N H^* W y \quad (2)$$

where we introduced the coefficient matrix

$$P_N = (\mu \lambda^{N+1} I + H^* W H)^{-1}. \quad (3)$$

We further let \hat{y} denote the vector

$$\hat{y} \triangleq H w_N. \quad (4)$$

We will refer to \hat{y} as the *regularized* projection (or simply projection) of the observation vector y onto the range space of H , which is written as $\mathcal{R}(H)$.

We also define two estimation error vectors: the *a posteriori* error vector

$$e_N = y - H w_N$$

and the *a priori* error vector

$$\epsilon_N = y - H w_{N-1}$$

where w_{N-1} is the solution to a least-squares problem similar to (1) with data up to time $N-1$ (and with $\mu \lambda^{N+1}$ replaced by $\mu \lambda^N$). The minimum cost of (1) will be denoted by $\xi(N)$, and it is given by¹

$$\xi(N) = y^* W e_N. \quad (5)$$

The last entries of e_N and ϵ_N are called the *a posteriori* and the *a priori* estimation errors at time N , and they are given by

$$e(N) = d(N) - u_N w_N, \quad \epsilon(N) = d(N) - u_N w_{N-1}.$$

¹We may note here that in the absence of regularization ($\mu = 0$), the expression for the minimum cost can also be expressed in the equivalent form $\xi(N) = e_N^* W e_N$. This follows from the orthogonality property $H^* W e_N = 0$. However, when regularization is present, we need to use (5) instead. This fact distinguishes the derivations we will provide for the updates of the minimum costs of the so-called forward and backward prediction problems from those that assume no regularization. More on this later.

They are both related by a conversion factor

$$e(N) = \gamma(N) \epsilon(N)$$

where

$$\gamma(N) = 1 - u_N P_N u_N^*.$$

The well-known RLS algorithm allows us to update w_N recursively as follows:

$$\gamma^{-1}(N) = 1 + \lambda^{-1} u_N P_{N-1} u_N^* \quad (6)$$

$$g_N = \lambda^{-1} P_{N-1} u_N^* \gamma(N) \quad (7)$$

$$w_N = w_{N-1} + g_N \epsilon(N) \quad (8)$$

$$P_N = \lambda^{-1} P_{N-1} - g_N \gamma^{-1}(N) g_N^* \quad (9)$$

with $w_{-1} = 0$ and $P_{-1} = \mu^{-1} I$. It also holds that $g_N = P_N u_N^*$.

III. ORDER-RECURSIVE RELATIONS

We now derive several order-recursive relations. As mentioned in the introduction, we re-emphasize that the presentation in this section has three contributions. First, the arguments do not assume shift structure. Second, the derivation introduces and singles out a variable whose update is affected by data structure. Third, the derivation explicitly incorporates regularization.

Before proceeding, we should remark that since in the remainder of this paper we deal primarily with order-recursive least-squares problems, it becomes important to explicitly indicate the size of all quantities involved (in addition to a time index). For example, we will write $w_{M,N}$ instead of w_N to indicate that it is a vector of order M that is computed by using data up to time N . We will also write $H_{M,N}$ instead of H to indicate that it is a matrix with row vectors of size M and with data up to time N . Similarly, we write y_N instead of y and W_N instead of W so that problem (1) becomes

$$\min_{w_M} [\mu \lambda^{N+1} \|w_M\|^2 + (y_N - H_{M,N} w_M)^* W_N (y_N - H_{M,N} w_M)]$$

and its solution is $w_{M,N}$. In a similar vein, we will write

$$\{\hat{y}_{M,N}, e_{M,N}, \epsilon_{M,N}, e_M(N), \epsilon_M(N), \gamma_M(N), \xi_M(N)\}.$$

A. Order Updating

Assume (for simplicity of presentation) that $M = 3$, and consider the data matrix

$$H_{3,N} = \begin{bmatrix} u(0,0) & u(0,1) & u(0,2) \\ u(1,0) & u(1,1) & u(1,2) \\ u(2,0) & u(2,1) & u(2,2) \\ \vdots & \vdots & \vdots \\ u(N,0) & u(N,1) & u(N,2) \end{bmatrix}.$$

²We may remark that without the factor λ^{N+1} in the cost (1), the above RLS recursions would not be accurate.

The (regularized) projection of y_N onto $\mathcal{R}(H_{3,N})$ is given by [cf. (4)]

$$\hat{y}_{3,N} = H_{3,N}P_{3,N}H_{3,N}^*W_N y_N.$$

Now, suppose that one more column is appended to $H_{3,N}$, i.e.,

$$H_{4,N} = [H_{3,N} \quad x_{3,N}] \quad (10)$$

where

$$x_{3,N} = \begin{bmatrix} u(0,3) \\ u(1,3) \\ u(2,3) \\ \vdots \\ u(N,3) \end{bmatrix}.$$

The (regularized) projection of y_N onto $\mathcal{R}(H_{4,N})$ is now

$$\hat{y}_{4,N} = H_{4,N}P_{4,N}H_{4,N}^*W_N y_N. \quad (11)$$

In order to relate both projections of the vector y_N , we note the following. The coefficient matrices $\{P_{3,N}, P_{4,N}\}$ are 3×3 and 4×4 , respectively, and they are defined by [cf. (3)]

$$P_{3,N}^{-1} = (\mu\lambda^{N+1}I + H_{3,N}^*W_N H_{3,N})$$

$$P_{4,N}^{-1} = (\mu\lambda^{N+1}I + H_{4,N}^*W_N H_{4,N}).$$

They are therefore related via

$$P_{4,N}^{-1} = \begin{bmatrix} P_{3,N}^{-1} & H_{3,N}^*W_N x_{3,N} \\ x_{3,N}^*W_N H_{3,N} & \mu\lambda^{N+1} + x_{3,N}^*W_N x_{3,N} \end{bmatrix}.$$

Inverting both sides, we get

$$P_{4,N} = \begin{bmatrix} P_{3,N} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\mu\lambda^{N+1} + \xi_3^b(N)} \cdot \begin{bmatrix} -w_{3,N}^b \\ 1 \end{bmatrix} \begin{bmatrix} -w_{3,N}^b & 1 \end{bmatrix} \quad (12)$$

where $w_{3,N}^b$ is the solution to the least-squares problem:

$$\min_{w_3^b} [\mu\lambda^{N+1} \|w_3^b\|^2 + (x_{3,N} - H_{3,N}w_3^b)^*W_N(x_{3,N} - H_{3,N}w_3^b)]$$

and $\xi_3^b(N)$ is the corresponding minimum cost. This problem projects $x_{3,N}$ onto $\mathcal{R}(H_{3,N})$. Let

$$b_{3,N} = x_{3,N} - H_{3,N}w_{3,N}^b$$

denote the resulting (backward) estimation error vector. Substituting (12) into (11), we find that the projections $\{\hat{y}_{4,N}, \hat{y}_{3,N}\}$ are related via

$$\hat{y}_{4,N} = \hat{y}_{3,N} + \frac{b_{3,N}^*W_N y_N}{\mu\lambda^{N+1} + \xi_3^b(N)} b_{3,N}. \quad (13)$$

Subtracting y_N from both sides, we obtain a relation between the corresponding a posteriori estimation error vectors

$$e_{4,N} = e_{3,N} - \kappa_3(N)b_{3,N} \quad (14)$$

where we define the scalar

$$\kappa_3(N) \triangleq \frac{b_{3,N}^*W_N y_N}{\mu\lambda^{N+1} + \xi_3^b(N)}$$

$$\triangleq \frac{\rho_3^*(N)}{\mu\lambda^{N+1} + \xi_3^b(N)}. \quad (15)$$

We therefore derived an order-update relation (14) for the *a posteriori* error vectors. The recursion however depends on $b_{3,N}$. We are thus motivated to study the propagation of $b_{3,N}$ more closely.

B. Backward Estimation Problem

We start by partitioning $H_{3,N}$ of (10) into

$$H_{3,N} = [x_{0,N} \quad \bar{H}_{2,N}]$$

so that $H_{4,N}$ is now partitioned as

$$H_{4,N} = [x_{0,N} \quad \bar{H}_{2,N} \quad x_{3,N}]. \quad (16)$$

Using arguments similar to those that led to the update equation (14) for $e_{4,N}$, it is straightforward to verify that $b_{3,N}$ can be obtained as follows:

$$b_{3,N} = \bar{b}_{2,N} - \kappa_2^b(N)f_{2,N}$$

where the scalar coefficient $\kappa_2^b(N)$ is defined as

$$\kappa_2^b(N) \triangleq \frac{f_{2,N}^*W_N x_{3,N}}{\mu\lambda^{N+1} + \xi_2^b(N)}$$

$$\triangleq \frac{\delta_2(N)}{\mu\lambda^{N+1} + \xi_2^b(N)} \quad (17)$$

and $f_{2,N}$ is the residual error that results from the solution of the least-squares problem

$$\min_{w_2^f} [\mu\lambda^{N+1} \|w_2^f\|^2 + (x_{0,N} - \bar{H}_{2,N}w_2^f)^*W_N(x_{0,N} - \bar{H}_{2,N}w_2^f)]$$

whose minimum cost we denote by $\xi_2^f(N)$. This problem projects $x_{0,N}$ onto $\mathcal{R}(\bar{H}_{2,N})$. Likewise, $\bar{b}_{2,N}$ is the residual error that results from the solution of the least-squares problem

$$\min_{w_2^{\bar{b}}} [\mu\lambda^{N+1} \|w_2^{\bar{b}}\|^2 + (x_{3,N} - \bar{H}_{2,N}w_2^{\bar{b}})^*W_N(x_{3,N} - \bar{H}_{2,N}w_2^{\bar{b}})]$$

whose minimum cost we denote by $\xi_2^{\bar{b}}(N)$. This problem projects $x_{3,N}$ onto $\mathcal{R}(\bar{H}_{2,N})$.

C. Forward Estimation Problem

By similar arguments, $f_{2,N}$ can be updated as follows:

$$f_{3,N} = f_{2,N} - \kappa_2^f(N)\bar{b}_{2,N}$$

where $\kappa_2^f(N)$ is defined as

$$\kappa_2^f(N) \triangleq \frac{\bar{b}_{2,N}^*W_N x_{0,N}}{\mu\lambda^{N+1} + \xi_2^f(N)}$$

$$= \frac{\delta_2^*(N)}{\mu\lambda^{N+1} + \xi_2^f(N)}. \quad (18)$$

Note that we used $\delta_2^*(N)$ in the numerator of $\kappa_2^f(N)$ and $\delta_2(N)$ in the numerator of $\kappa_2^b(N)$ in (17) since it can be easily verified that

$$[f_{2,N}^* W_N x_{3,N}]^* = \bar{b}_{2,N}^* W_N x_{0,N}. \quad (19)$$

Summarizing, we have so far derived the following order-update relations for the error vectors $\{c_{M,N}, b_{M,N}, f_{M,N}\}$ (which are written here for a generic order M):

$$\begin{cases} c_{M+1,N} = c_{M,N} - \kappa_M(N) b_{M,N} \\ b_{M+1,N} = \bar{b}_{M,N} - \kappa_M^b(N) f_{M,N} \\ f_{M+1,N} = f_{M,N} - \kappa_M^f(N) \bar{b}_{M,N}. \end{cases}$$

We still need to derive a relation for $\bar{b}_{M,N}$. We postpone this discussion to Section III-G due to its intrinsic dependence on data structure.

If we extract the last entries of the above vectors, we obtain the following relations:

$$\begin{cases} c_{M+1}(N) = c_M(N) - \kappa_M(N) b_M(N) \\ b_{M+1}(N) = \bar{b}_M(N) - \kappa_M^b(N) f_M(N) \\ f_{M+1}(N) = f_M(N) - \kappa_M^f(N) \bar{b}_M(N) \end{cases}$$

where

$$\begin{cases} \kappa_M(N) = \frac{\rho_M^*(N)}{\mu\lambda^{N+1} + \xi_M^b(N)} \\ \kappa_M^b(N) = \frac{\delta_M(N)}{\mu\lambda^{N+1} + \xi_M^f(N)} \\ \kappa_M^f(N) = \frac{\delta_M^*(N)}{\mu\lambda^{N+1} + \xi_M^b(N)}. \end{cases}$$

We now show how to update the quantities

$$\{\delta_M(N), \rho_M(N), \xi_M^b(N), \xi_M^f(N), \xi_M^*(N)\}$$

which are needed in the evaluation of the (reflection) coefficients $\{\kappa_M(N), \kappa_M^b(N), \kappa_M^f(N)\}$. To do so, we first derive below a general update result. It is important to re-emphasize that this result will be independent of any data structure in the matrix $H_{M,N}$ (cf. [20]).

D. General Time-Update Result

Consider a generic data matrix of the form

$$\begin{bmatrix} x & \bar{H} & z \end{bmatrix} \quad (20)$$

where x and z are column vectors, and \bar{H} is a matrix of appropriate dimensions. Define the weighted inner product

$$\Delta \triangleq x^* W \tilde{z}$$

where \tilde{z} is the residual vector from a regularized projection of z onto $\mathcal{R}(\bar{H})$, namely, $\tilde{z} = z - \bar{H}w_z$, where w_z is obtained by solving

$$\min_w [\mu\lambda^{N+1} \|w\|^2 + (z - \bar{H}w)^* W (z - \bar{H}w)] \quad (21)$$

where, as before, $W = \text{diag}\{\lambda^N, \dots, \lambda, 1\}$.

Now, assume that one more row is appended to the matrix (20), say

$$\begin{bmatrix} x & \bar{H} & z \\ \alpha & h & \beta \end{bmatrix} \triangleq \begin{bmatrix} x_1 & \bar{H}_1 & z_1 \end{bmatrix}$$

and introduce the corresponding factor $\Delta_1 = x_1^* W_1 \tilde{z}_1$, where $W_1 = (\lambda W \oplus 1)$. We would like to relate Δ_1 and Δ (i.e., we would like to determine an order-update relation for Δ).

As above, let w_{z_1} denote the solution of a problem similar to (21) with $\{z, \bar{H}, W, \lambda^{N+1}\}$ replaced by $\{z_1, \bar{H}_1, W_1, \lambda^{N+2}\}$. Likewise, let w_{x_1} denote the solution of a problem similar to (21) with $\{z, \bar{H}, W, \lambda^{N+1}\}$ replaced by $\{x_1, \bar{H}_1, W_1, \lambda^{N+2}\}$. Now, define the *a posteriori* errors

$$\tilde{\alpha} = \alpha - h w_{x_1}, \quad \tilde{\beta} = \beta - h w_{z_1}$$

and the conversion factor

$$\gamma = 1 - h[\mu\lambda^{N+2} I + \bar{H}_1^* W_1 \bar{H}_1]^{-1} h^* \triangleq 1 - h \bar{P}_1 h^*.$$

From the definition of Δ_1 , we have

$$\begin{aligned} \Delta_1 &= \begin{bmatrix} x^* & \alpha^* \end{bmatrix} \begin{bmatrix} \lambda W & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} z \\ \beta \end{bmatrix} - \begin{bmatrix} \bar{H} \\ h \end{bmatrix} w_{z_1} \right) \\ &= \lambda x^* W z + \alpha^* \beta - (\lambda x^* W \bar{H} + \alpha^* h) w_{z_1}. \end{aligned} \quad (22)$$

The RLS recursion (6) allows us to relate w_{z_1} and w_z as

$$w_{z_1} = w_z + \gamma^{-1} \bar{P}_1 h^* \tilde{\beta}.$$

Substituting into (22), we obtain, after grouping terms

$$\boxed{\Delta_1 = \lambda \Delta + \frac{\tilde{\alpha}^* \tilde{\beta}}{\gamma}} \quad (23)$$

E. Time-Update Relations

We can now use the general result (23) to derive updates for

$$\{\delta_M(N), \rho_M(N), \xi_M^b(N), \xi_M^f(N), \xi_M^*(N)\}.$$

Consider, for example, the quantity

$$\delta_M(N) = x_{0,N}^* W_N \bar{b}_{M,N}$$

which appears in the numerator of $\kappa_M^f(N)$ [see (18) for $M = 2$]. We can partition $H_{M+2,N}$ as

$$\begin{aligned} H_{M+2,N} &= \begin{bmatrix} x_{0,N} & \bar{H}_{M,N} & x_{M+1,N} \\ x_{0,N-1} & \bar{H}_{M,N-1} & x_{M+1,N-1} \\ u(N,0) & \bar{u}_{M,N} & u(N,M+1) \end{bmatrix} \end{aligned}$$

and make the identifications $x_1 \rightarrow x_{0,N}$, $\tilde{z}_1 \rightarrow \bar{b}_{M,N}$, $\bar{H}_1 = \bar{H}_{M,N}$, and $W_1 \rightarrow W_N$. It then follows that

$$\boxed{\delta_M(N) = \lambda \delta_M(N-1) + \frac{f_M^*(N) \bar{b}_M(N)}{\bar{\gamma}_M(N)}} \quad (24)$$

where the conversion factor $\bar{\gamma}_M(N)$ in this case is defined by

$$\begin{aligned}\bar{\gamma}_M(N) &= 1 - \bar{u}_{M,N} \bar{P}_{M,N} \bar{u}_{M,N}^*, \\ \bar{P}_{M,N} &= [\mu \lambda^{N+1} I + \bar{H}_{M,N}^* W_N \bar{H}_{M,N}]^{-1}.\end{aligned}$$

In a similar vein, we can justify the following time-update equations:

$$\rho_M(N) = \lambda \rho_M(N-1) + \frac{e_M^*(N) b_M(N)}{\gamma_M(N)} \quad (25)$$

$$\xi_M^{\bar{b}}(N) = \lambda \xi_M^{\bar{b}}(N-1) + \frac{|\bar{b}_M(N)|^2}{\bar{\gamma}_M(N)} \quad (26)$$

$$\xi_M^f(N) = \lambda \xi_M^f(N-1) + \frac{|f_M(N)|^2}{\bar{\gamma}_M(N)} \quad (27)$$

$$\xi_M^b(N) = \lambda \xi_M^b(N-1) + \frac{|b_M(N)|^2}{\gamma_M(N)} \quad (28)$$

where the conversion factor $\gamma_M(N)$ is defined by

$$\begin{aligned}\gamma_M(N) &= 1 - u_{M,N} P_{M,N} u_{M,N}^*, \\ P_{M,N} &= [\mu \lambda^{N+1} I + H_{M,N}^* W_N H_{M,N}]^{-1}.\end{aligned}$$

F. Order-Update Relations

We can order update the conversion factors as follows. Note that the last row of $H_{M+1,N}$ can be partitioned as

$$u_{M+1,N} = [u_{M,N} \quad u(N, M)]$$

so that by multiplying a relation of the form (12) for $P_{M+1,N}$ by $u_{M+1,N}$ from the left and by its conjugate transpose from the right, we get

$$\boxed{\gamma_{M+1}(N) = \gamma_M(N) - \frac{|b_M(N)|^2}{\mu \lambda^{N+1} + \xi_M^b(N)}} \quad (29)$$

Similarly, we get

$$\boxed{\bar{\gamma}_{M+1}(N) = \bar{\gamma}_M(N) - \frac{|\bar{b}_M(N)|^2}{\mu \lambda^{N+1} + \xi_M^{\bar{b}}(N)}} \quad (30)$$

In addition, using the order-update relations for the vectors $b_{M+1,N}$ and $f_{M+1,N}$ and the defining relations

$$\begin{aligned}\xi_{M+1}^f(N) &= x_{0,N}^* W_N f_{M+1,N} \\ \xi_{M+1}^b(N) &= x_{M+1,N}^* W_N b_{M+1,N}\end{aligned}$$

we obtain the following order recursions for the minimum costs $\xi_M^b(N)$ and $\xi_M^f(N)$:

$$\begin{aligned}\xi_{M+1}^f(N) &= \xi_M^f(N) - \frac{|\delta_M(N)|^2}{\mu \lambda^{N+1} + \xi_M^{\bar{b}}(N)} \\ \xi_{M+1}^b(N) &= \xi_M^{\bar{b}}(N) - \frac{|\delta_M(N)|^2}{\mu \lambda^{N+1} + \xi_M^f(N)}.\end{aligned}$$

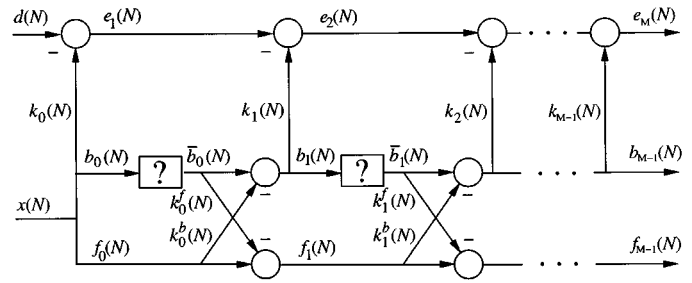


Fig. 1. General RLS lattice network structure.

G. Significance of Data Structure

Thus far, we have derived almost all the necessary recursions for the development of an adaptive lattice filter. All the results hold for *arbitrary* data structures. The only update missing is the one for the error sequence $\{\bar{b}_M(N)\}$. As is shown schematically in Fig. 1 by the boxes with question marks, we need to know how to generate the errors $\{\bar{b}_M(N)\}$. It is the update of these variables that is directly affected by data structure and it is the key to achieving a fast algorithm [by fast, we mean $\mathcal{O}(M)$ operations per iteration for a filter of order M].

To illustrate this point, consider, as an example, the case of prewindowed input data with shift structure, e.g., for $M = 3$

$$H_{4,N} = \begin{bmatrix} u(0) & 0 & 0 & 0 \\ u(1) & u(0) & 0 & 0 \\ u(2) & u(1) & u(0) & 0 \\ u(3) & u(2) & u(1) & u(0) \\ \vdots & \vdots & \vdots & \vdots \\ u(N) & u(N-1) & u(N-2) & u(N-3) \end{bmatrix}.$$

Then, any two successive columns $\{x_{i,N}, x_{i+1,N}\}$ of $H_{M,N}$ are related by the lower triangular shift matrix Z , i.e.,

$$x_{i+1,N} = Z x_{i,N} \quad (31)$$

so that the following always holds for all M :

$$\bar{H}_{M,N} = Z H_{M,N}, \quad Z H_{M,N} = \begin{bmatrix} 0 \\ H_{M,N-1} \end{bmatrix}.$$

Using these relations in the definitions of $\bar{b}_{M,N}$ and $b_{M,N}$, viz., in

$$\begin{cases} \bar{b}_{M,N} = x_{M+1,N} - \bar{H}_{M,N} w_{M,N}^{\bar{b}} \\ b_{M,N} = x_{M,N} - H_{M,N} w_{M,N}^b \end{cases} \quad (32)$$

we can easily verify that

$$\bar{b}_{M,N} = \begin{bmatrix} 0 \\ b_{M,N-1} \end{bmatrix}$$

and hence

$$\bar{b}_M(N) = b_M(N-1)$$

which is a widely known result. In a similar vein, it will also hold that

$$\xi_M^{\bar{b}}(N) = \xi_M^b(N-1) \quad \text{and} \quad \bar{\gamma}_M(N) = \gamma_M(N-1).$$

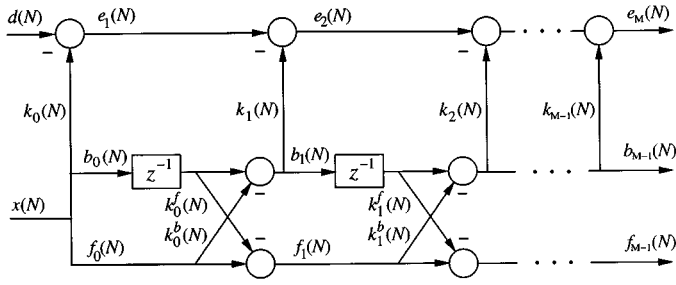


Fig. 2. RLS lattice network for the case of shift data structure.

These equalities eliminate the need for recursions (26) and (30), and the general lattice recursions of this paper collapse to the well-known tapped-delay-line lattice network depicted in Fig. 2. The corresponding regularized lattice equations are listed in Table I.

We may note that we have redefined the minimum cost variables in order to save addition operations. For example, we defined

$$\begin{aligned}\zeta_M^b(N) &\triangleq \mu\lambda^{N+1} + \xi_M^b(N) \\ \zeta_M^f(N) &\triangleq \mu\lambda^{N+1} + \xi_M^f(N).\end{aligned}$$

Then, it is easy to verify that these new quantities satisfy recursions similar to those of $\{\zeta_M^b(N), \xi_M^f(N)\}$, namely

$$\begin{aligned}\zeta_M^f(N) &= \lambda\zeta_M^f(N-1) + \frac{|f_M(N)|^2}{\gamma_M(N-1)} \\ \zeta_M^b(N) &= \lambda\zeta_M^b(N-1) + \frac{|b_M(N)|^2}{\gamma_M(N)} \\ \zeta_{M+1}^f(N) &= \zeta_M^f(N) - \frac{|\delta_M(N)|^2}{\zeta_M^b(N)} \\ \zeta_{M+1}^b(N) &= \zeta_M^b(N) - \frac{|\delta_M(N)|^2}{\zeta_M^f(N)}\end{aligned}$$

but with the initial conditions $\zeta_M^b(-1) = \zeta_M^f(-1) = \mu$.

Observe that the variables $\{\zeta_M^b(N), \zeta_M^f(N)\}$ do not correspond to the exact values of the minimum costs for the backward and forward prediction problems. Only when $\lambda < 1$ and $N \rightarrow \infty$, they tend to coincide with the actual values $\{\xi_M^b(N), \xi_M^f(N)\}$.³

Now, what if two successive columns of the input data matrix $H_{M,N}$ are not shifted versions of each other as in (31) but are instead related by a more general matrix Φ ? Would it still be possible to derive a fast lattice algorithm? Interesting enough, the answer is positive. We demonstrate this fact in the next section by considering an important example. The result will show that it is possible to move beyond what has been developed so far in the literature for shift-structured data and to develop *exact* RLS lattice algorithms for more general data.

³This point serves as a good example of one of the issues we raised earlier, namely, by solving regularized problems from the start, we can clarify the significance and the meaning of the different variables that appear in the lattice recursions.

 TABLE I
 CLASSICAL RLS-FIR ADAPTIVE LATTICE ALGORITHM

Initialization

For $M = 0$ *to* $M - 1$ *set:*

μ is a small positive number.
 $\delta_M(-1) = \rho_M(-1) = 0$
 $\zeta_M^f(-1) = \zeta_M^b(-1) = \mu$

For $N \geq 0$, *repeat:*

$\gamma_0(N) = 1$
 $b_0(N) = f_0(N) = u(N)$

For $M = 0$ *to* $M - 1$, *repeat:*

$$\begin{aligned}\zeta_M^f(N) &= \lambda\zeta_M^f(N-1) + \frac{|f_M(N)|^2}{\gamma_M(N-1)} \\ \zeta_M^b(N) &= \lambda\zeta_M^b(N-1) + \frac{|b_M(N)|^2}{\gamma_M(N)} \\ \delta_M(N) &= \lambda\delta_M(N-1) + \frac{f_M^*(N)b_M(N-1)}{\gamma_M(N-1)} \\ \rho_M(N) &= \lambda\rho_M(N-1) + \frac{e_M^*(N)b_M(N)}{\gamma_M(N)}\end{aligned}$$

$$\gamma_{M+1}(N) = \gamma_M(N) - \frac{|b_M(N)|^2}{\zeta_M^b(N)}$$

$$\begin{aligned}\kappa_M^b(N) &= \frac{\delta_M(N)}{\zeta_M^b(N)} \\ \kappa_M^f(N) &= \frac{\rho_M(N)}{\zeta_M^b(N-1)} \\ \kappa_M(N) &= \frac{\rho_M(N)}{\zeta_M^b(N)}\end{aligned}$$

$$\begin{aligned}e_{M+1}(N) &= e_M(N) - \kappa_M(N)b_M(N) \\ b_{M+1}(N) &= b_M(N-1) - \kappa_M^b(N)f_M(N) \\ f_{M+1}(N) &= f_M(N) - \kappa_M^f(N)b_M(N-1)\end{aligned}$$

IV. RLS LAGUERRE ADAPTIVE FILTERING

Consider the Laguerre-based model of Fig. 3, where

$$L_0(z) = \frac{\sqrt{1-a^2}}{1-az^{-1}} \quad \text{and} \quad L(z) = \frac{z^{-1}-a}{1-az^{-1}}, \quad 0 < |a| < 1. \quad (33)$$

Note that $L(z)$ is a first-order allpass system and that, unlike a general IIR structure, the poles of the Laguerre-based model are fixed at a .⁴ (The choice of a can be optimized offline according to some criterion; see, e.g., [21].) The input to the Laguerre filter at time N is denoted by $s(N)$, and the coefficients that combine the outputs of the successive sections $\{L_0(z), L(z)\}$ are denoted by $\{w_k\}$.

Now, consider the case of prewindowed input data (i.e., $s(i) = 0$ for $i \leq 0$ and zero initial conditions). Using the difference equations that define $\{L_0(z), L(z)\}$, it is possible to relate two successive columns of the data matrix in this case as

$$x_{i+1,N} = \Phi_N x_{i,N}$$

⁴The case of complex poles can also be handled. For simplicity, we assume real a .

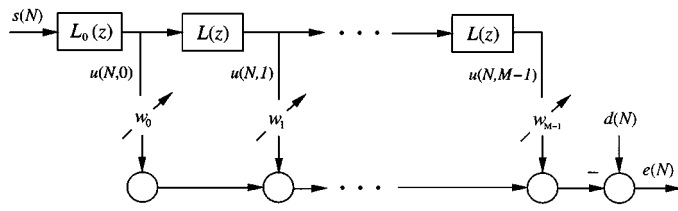


Fig. 3. Transversal Laguerre structure for adaptive filtering.

where Φ_N is an $(N + 1) \times (N + 1)$ lower triangular Toeplitz matrix of the form

$$\Phi_N = \begin{bmatrix} -a & & & & \\ 1 - a^2 & -a & & & \\ a(1 - a^2) & 1 - a^2 & -a & & \\ \vdots & \vdots & \ddots & \ddots & \\ a^{(N-1)}(1 - a^2) & \dots & \dots & 1 - a^2 & -a \end{bmatrix}. \quad (34)$$

Of course, it also holds that

$$\bar{H}_{M,N} = \Phi_N H_{M,N}. \quad (35)$$

A. Exploiting Data Structure

Now, referring back to the definitions of the error vectors $\{b_{M,N}, \bar{b}_{M,N}\}$ in (32), we see that we need to relate $\{w_{M,N}^b, w_{M,N}^{\bar{b}}\}$. These vectors are given by

$$\begin{aligned} w_{M,N}^b &= P_{M,N} H_{M,N}^* W_N x_{M,N} \\ w_{M,N}^{\bar{b}} &= \bar{P}_{M,N} \bar{H}_{M,N}^* W_N x_{M+1,N} \\ P_{M,N} &= [\mu \lambda^{N+1} I + H_{M,N}^* W_N H_{M,N}]^{-1} \\ \bar{P}_{M,N} &= [\mu \lambda^{N+1} I + \bar{H}_{M,N}^* W_N \bar{H}_{M,N}]^{-1}. \end{aligned}$$

For simplicity of notation, we are going to write $\{H, P, \Phi, W\}$ instead of $\{H_{M,N}, P_{M,N}, \Phi_N, W_N\}$ in the following derivation. Later, we will state the results with the correct subscripts. We will also assume in the sequel that $\lambda = 1$ so that $W = I$ ⁵

Using (35), we get

$$w_{M,N}^{\bar{b}} = [\mu I + H^* \Phi^* \Phi H]^{-1} H^* \Phi^* \Phi x_{M,N}. \quad (36)$$

Now note that $\Phi^* \Phi$ is a rank-one modification of the identity matrix, namely, it satisfies

$$\Phi_N^* \Phi_N = I - c_N c_N^* \quad (37)$$

where

$$c_N \triangleq \sqrt{1 - a^2} [a^N \quad a^{N-1} \quad \dots \quad a \quad 1]^T.$$

[We will also write c instead of c_N for simplicity.]

Substituting (37) into (36), we obtain

$$\begin{aligned} w_{M,N}^{\bar{b}} &= [\mu I + H^* (I - c c^*) H]^{-1} H^* (I - c c^*) x_{M,N} \\ &= (P^{-1} - H^* c c^* H)^{-1} H^* (I - c c^*) x_{M,N}. \end{aligned} \quad (38)$$

⁵For more general W and Φ , the arguments of this section would require the difference $W - \Phi^* W \Phi$ to have low rank.

For simplicity, define $v \triangleq c^* H$. Then, using the matrix inversion lemma, we get

$$(P^{-1} - v^* v)^{-1} = P - \frac{P v^* v P}{v P v^* - 1}.$$

Substituting this expression into (38) leads to a desired relation between $\{w_{M,N}^b, w_{M,N}^{\bar{b}}\}$:

$$\begin{aligned} w_{M,N}^{\bar{b}} &= P \left(H^* - \frac{v^* v P H^*}{v P v^* - 1} - v^* c^* + \frac{v^* v P v^* c^*}{v P v^* - 1} \right) x_{M,N} \\ &= w_{M,N}^b - \frac{P H^* c c^*}{1 - c^* H P H^* c} b_{M,N}. \end{aligned}$$

If we now multiply by $\bar{H}_{M,N}$ from the left and subtract $x_{M+1,N}$, then using (32) and (35), we obtain

$$\bar{b}_{M,N} = \Phi b_{M,N} + \frac{c^* b_{M,N}}{1 - c^* \hat{c}_{M,N}} \Phi \hat{c}_{M,N} \quad (39)$$

where we defined the vector

$$\hat{c}_{M,N} \triangleq H_{M,N} P_{M,N} H_{M,N}^* c_N.$$

This vector has the interpretation of being the regularized projection of c_N onto $\mathcal{R}(H_{M,N})$.

In view of the above, we find that the last entry of $\bar{b}_{M,N}$ is given by

$$\bar{b}_M(N) = \phi_N b_{M,N} + \frac{c_N^* b_{M,N}}{1 - c_N^* \hat{c}_{M,N}} \phi_N \hat{c}_{M,N} \quad (40)$$

where ϕ_N is the last row of Φ_N . The above relation involves four growing-length inner products on the right-hand side:

$$\{\phi_N b_{M,N}, c_N^* b_{M,N}, c_N^* \hat{c}_{M,N}, \phi_N \hat{c}_{M,N}\}.$$

We will show that the first two are related to each other, whereas the last two are also related to each other. This will follow as a result of the fact that ϕ_N and c_N have similar forms. In this way, we shall need only to develop order-recursive updates for two of these inner-product terms.

To this end, first note that we can simplify (40) by exploiting the similarity between the vectors ϕ_N and c_N , viz.

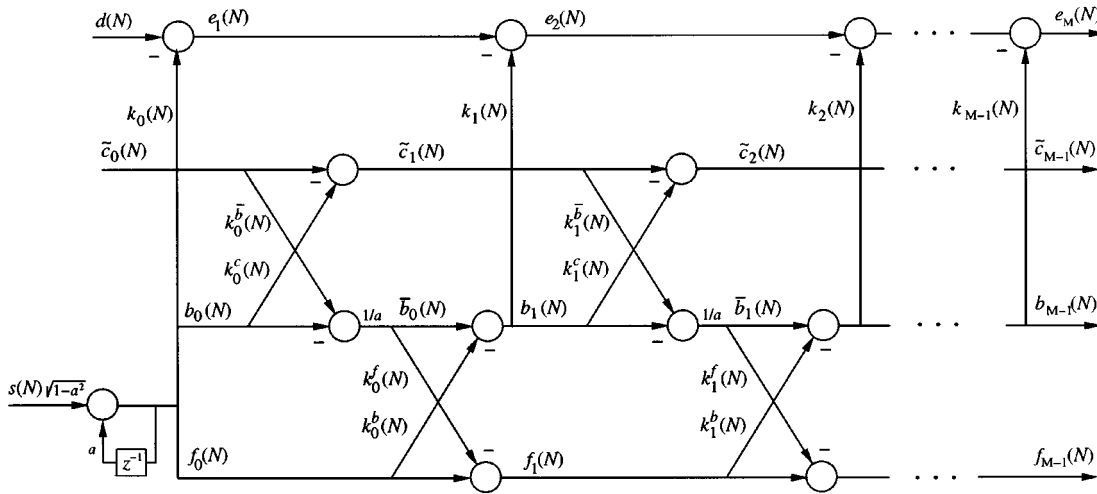
$$\phi_N = \frac{\sqrt{1 - a^2}}{a} c_N^* + \begin{bmatrix} 0 & \dots & 0 & -\frac{1}{a} \end{bmatrix}$$

to write

$$\begin{aligned} \phi_N b_{M,N} &= \frac{\sqrt{1 - a^2}}{a} c_N^* b_{M,N} - \frac{b_M(N)}{a} \\ \phi_N \hat{c}_{M,N} &= \frac{\sqrt{1 - a^2}}{a} c_N^* \hat{c}_{M,N} - \frac{\hat{c}_M(N)}{a}. \end{aligned}$$

Substituting these expressions into (40), we obtain, after some manipulations

$$\bar{b}_M(N) = \frac{1}{a} \left(\frac{c_N^* b_{M,N} \hat{c}_M(N)}{1 - c_N^* \hat{c}_{M,N}} - b_M(N) \right)$$


 Fig. 4. RLS-Laguerre lattice filter, where $\bar{c}_0(N) = \sqrt{1-a^2}$.

which contains only two inner products

$$\tau_M(N) \triangleq c_N^* b_{M,N}, \quad \zeta_M^c(N) \triangleq 1 - c_N^* \hat{c}_{M,N}.$$

With these definitions, we can write

$$\bar{b}_M(N) = -\frac{1}{a} \left(b_M(N) - \kappa_M^b(N) \tilde{c}_M(N) \right)$$

where we defined

$$\kappa_M^b(N) = \frac{\tau_M(N)}{\zeta_M^c(N)}$$

and $\tilde{c}_M(N) = c_M(N) - \hat{c}_M(N)$. [Here, $c_M(N)$ and $\hat{c}_M(N)$ denote the last entries of $c_{M,N}$ and $\hat{c}_{M,N}$, respectively.]

Hence, all we really need to know is how to update the quantity $\tilde{c}_M(N)$ and the inner products $\tau_M(N)$ and $\zeta_M(N)$.

First, note that $\hat{c}_{M,N}$ and $\tilde{c}_{M,N}$ can be order updated, in the same fashion as in (13) and (14), namely

$$\begin{aligned} \hat{c}_{M+1,N} &= \hat{c}_{M,N} + \frac{b_{M,N} \tau_M^*(N)}{\mu + \xi_M^b(N)} \\ \tilde{c}_{M+1}(N) &= \tilde{c}_M(N) - \frac{b_M(N) \tau_M^*(N)}{\mu + \xi_M^b(N)} \\ &\triangleq \tilde{c}_M(N) - \kappa_M^c(N) b_M(N) \end{aligned}$$

where we defined

$$\kappa_M^c(N) \triangleq \frac{\tau_M^*(N)}{\mu + \xi_M^b(N)}.$$

In addition, multiplying the above recursion for $\hat{c}_{M,N}$ by c_N^* from the left, we obtain

$$c_N^* \hat{c}_{M+1,N} = c_N^* \hat{c}_{M,N} + \frac{c_N^* b_{M,N} b_{M,N}^* c_N}{\mu + \xi_M^b(N)}$$

and subtracting one from both sides, we get the following order-update recursion for $\zeta_M(N)$:

$$\zeta_{M+1}^c(N) = \zeta_M^c(N) - \frac{|\tau_M(N)|^2}{\zeta_M^b(N)}$$

From (39), we can derive an alternative recursion for $\xi_M^b(N)$. To see this, we multiply (39) by $x_{M+1,N}^* = x_{M,N}^* \Phi^*$ from the left and get

$$\xi_M^b(N) = x_{M,N}^* \Phi^* \Phi b_{M,N} + \kappa_M^b(N) x_{M,N}^* \Phi^* \Phi \hat{c}_{M,N}.$$

Now, using (37) and the substitutions $x_{M,N}^* b_{M,N} = \xi_M^b(N)$, $c_N^* b_{M,N} = \tau_M(N)$, and $c_N^* \hat{c}_{M,N} = 1 - \zeta_M^c(N)$, we get

$$\xi_M^b(N) = \xi_M^b(N) - \frac{\tau_M(N)}{\zeta_M^c(N)} x_{M,N}^* \hat{c}_{M,N}.$$

However, similar to (19), it holds that $x_{M,N}^* \tilde{c}_{M,N} = b_{M,N}^* c_N$ so that we arrive at the recursion

$$\xi_M^b(N) = \xi_M^b(N) - \frac{|\tau_M(N)|^2}{\zeta_M^c(N)}$$

B. Final Time-Update Recursions

Finally, it only remains to determine a recursive relation for $\tau_M(N)$. This step requires more effort. We start by noting that we can write $\tau_M(N)$ in the form

$$\tau_M(N) = \sqrt{1-a^2} [1 \ 1 \ \dots \ 1] A_N b_{M,N}$$

where $A_N = \text{diag}\{a^N, a^{N-1}, \dots, a, 1\}$. Observe that in so doing, we expressed $\tau_M(N)$ as the product of a constant vector by a diagonal weighting matrix A_N . Now, in order to update $\tau_M(N)$ to $\tau_{M+1}(N)$, we need to generalize our earlier result on the update of Δ to Δ_1 in Section III-D.

Thus, consider again a generic data matrix of the form

$$[x \ \bar{H} \ z]$$

and introduce the weighted inner product $\Delta = x^* \Lambda \tilde{z}$, for some positive-definite diagonal matrix Λ . Here, \tilde{z} denotes the residual vector from a regularized projection of z onto \overline{H} , namely, $\tilde{z} = z - \overline{H}w_z$, where w_z is obtained by solving

$$\min_w [\mu \lambda^{N+1} \|w\|^2 + (z - \overline{H}w)^* W (z - \overline{H}w)] \quad (41)$$

where, as before, $W = \text{diag}\{\lambda^N, \dots, \lambda, 1\}$. Note that now, however, the definition of Δ involves a center matrix Λ that is distinct from W . Now, consider the extended matrix

$$\begin{bmatrix} x & \overline{H} & z \\ \alpha & h & \beta \end{bmatrix} \triangleq [x_1 \quad \overline{H}_1 \quad z_1]$$

and introduce the corresponding factor $\Delta_1 = x_1^* \Lambda_1 \tilde{z}_1$, where Λ_1 is related to Λ via $\Lambda_1 = (\eta \Lambda \oplus 1)$, for some η . We again would like to relate Δ_1 and Δ (i.e., we would like to determine an order-update relation for Δ).

As above, let w_{z_1} denote the solution of a problem similar to (41) with $\{z, \overline{H}, W, \lambda^{N+1}\}$ replaced by $\{z_1, \overline{H}_1, (\lambda W \oplus 1), \lambda^{N+2}\}$. Likewise, let w_{x_1} denote the solution of a problem similar to (41) with $\{z, \overline{H}, W, \lambda^{N+1}\}$ replaced by $\{x_1, \overline{H}_1, (\lambda W \oplus 1), \lambda^{N+2}\}$. In addition, define the *a posteriori* error

$$\tilde{\beta} = \beta - h w_{z_1}.$$

Then, an argument similar to that in Section III-D will show that

$$\Delta_1 = \eta \Delta + \left[\alpha - \frac{\eta}{\lambda} h P \overline{H}^* \Lambda x \right]^* \tilde{\beta} \quad (42)$$

where P is defined by

$$P = \left(\mu \lambda^{N+1} + \overline{H}^* W \overline{H} \right)^{-1}.$$

Returning to the update of $\tau_M(N)$, we can now make the following identifications:

$$\begin{aligned} x_1 &\rightarrow \sqrt{1-a^2} [1 \quad 1 \quad \dots \quad 1]^T \\ \alpha &= \sqrt{1-a^2}, \quad \eta = a, \quad \lambda = 1 \\ \tilde{z}_1 &\rightarrow b_{M,N}, \quad W_1 = I, \quad \Lambda_1 \rightarrow A_N \\ \eta \Lambda x &\rightarrow \sqrt{1-a^2} [a^N \quad a^{N-1} \quad \dots \quad a]^T \end{aligned}$$

that is, $\eta \Lambda x$ corresponds to the top entries of c_N . In this case, the term $\eta h P \overline{H}^* \Lambda x$ would correspond to the (regularized) estimate of α that is based on the prior data in \overline{H} . Then, the difference $\alpha - \eta h P \overline{H}^* \Lambda x$ becomes equal to the *a priori* error in estimating α , which can be transformed to the *a posteriori* error $\tilde{c}_M(N)$ by means of the conversion factor $\gamma_M(N)$. This leads to the desired update equation

$$\tau_M(N+1) = a \tau_M(N) + \frac{\tilde{c}_M^*(N) b_M(N)}{\gamma_M(N)}$$

TABLE II
O(M) RLS-LAGUERRE LATTICE FILTER

Initialization	
<i>For M = 0 to M - 1 set:</i>	
μ is a small positive number.	
$\delta_M(-1) = \rho_M(-1) = \tau_M(-1) = 0$	
$\zeta_M^f(-1) = \zeta_M^b(-1) = \zeta_M^{\tilde{c}}(-1) = \mu$	
$\zeta_M^c(-1) = 1$	
<i>For N ≥ 0, repeat:</i>	
$u(N) = a u(N-1) + \sqrt{1-a^2} s(N)$	
$\gamma_0(N) = 1 \quad e_0(N) = d(N)$	
$\tilde{\gamma}_0(N) = 1 \quad f_0(N) = u(N)$	
$\tilde{c}_0(N) = \sqrt{1-a^2} \quad b_0(N) = u(N)$	
<i>For M = 0 to M - 1, repeat:</i>	
$\tau_M(N) = a \tau_M(N-1) + \frac{\tilde{c}_M^*(N) b_M(N)}{\gamma_M(N)}$	
$\zeta_M^c(N) = a^2 \zeta_M^c(N-1) + \frac{ \tilde{e}_M(N) ^2}{\gamma_M(N)}$	
$\kappa_M^b(N) = \frac{\tau_M(N)}{\zeta_M^c(N)}$	
$\tilde{b}_M(N) = -\frac{1}{a} (b_M(N) - \kappa_M^b(N) \tilde{c}_M(N))$	
$\zeta_M^f(N) = \zeta_M^f(N-1) + \frac{ f_M(N) ^2}{\tilde{\gamma}_M(N)}$	
$\zeta_M^b(N) = \zeta_M^b(N-1) + \frac{ \tilde{b}_M(N) ^2}{\gamma_M(N)}$	
$\zeta_M^{\tilde{c}}(N) = \zeta_M^{\tilde{c}}(N-1) + \frac{ \tilde{e}_M(N) ^2}{\tilde{\gamma}_M(N)}$	
$\delta_M(N) = \delta_M(N-1) + \frac{f_M^*(N) \tilde{b}_M(N)}{\tilde{\gamma}_M(N)}$	
$\rho_M(N) = \rho_M(N-1) + \frac{e_M^*(N) \tilde{b}_M(N)}{\tilde{\gamma}_M(N)}$	
$\gamma_{M+1}(N) = \gamma_M(N) - \frac{ \tilde{b}_M(N) ^2}{\zeta_M^b(N)}$	
$\tilde{\gamma}_{M+1}(N) = \tilde{\gamma}_M(N) - \frac{ \tilde{e}_M(N) ^2}{\zeta_M^{\tilde{c}}(N)}$	
$\kappa_M^c(N) = \frac{\tau_M(N)}{\zeta_M^c(N)}$	
$\kappa_M^b(N) = \frac{\tilde{b}_M(N)}{\zeta_M^b(N)}$	
$\kappa_M^f(N) = \frac{\delta_M(N)}{\zeta_M^f(N)}$	
$\kappa_M(N) = \frac{\rho_M(N)}{\zeta_M^{\tilde{c}}(N)}$	
$\tilde{c}_{M+1}(N) = \tilde{c}_M(N) - \kappa_M^c(N) b_M(N)$	
$e_{M+1}(N) = e_M(N) - \kappa_M(N) b_M(N)$	
$b_{M+1}(N) = \tilde{b}_M(N) - \kappa_M^b(N) f_M(N)$	
$f_{M+1}(N) = f_M(N) - \kappa_M^f(N) \tilde{b}_M(N)$	
Alternative order-updates:	
$\zeta_{M+1}^f(N) = \zeta_M^f(N) - \frac{ \tilde{b}_M(N) ^2}{\zeta_M^b(N)}$	
$\zeta_{M+1}^b(N) = \zeta_M^b(N) - \frac{ \tilde{e}_M(N) ^2}{\zeta_M^{\tilde{c}}(N)}$	
$\zeta_{M+1}^{\tilde{c}}(N) = \zeta_M^{\tilde{c}}(N) - \frac{ \tau_M(N) ^2}{\zeta_M^c(N)}$	
$\zeta_M^{\tilde{c}}(N) = \zeta_M^{\tilde{c}}(N) - \frac{ \tau_M(N) ^2}{\zeta_M^c(N)}$	

Using similar arguments and the fact that

$$\zeta_M^c(N) = [1 - c_N^* c_N] - c_N^* \tilde{c}_{M,N} = a^{2(N+1)} - c_N^* \tilde{c}_{M,N}$$

TABLE III
COMPARISON OF THE COMPUTATIONAL COST OF AN RLS-LAGUERRE LATTICE FILTER OF ORDER M WITH THAT OF AN RLS-FIR LATTICE FILTER OF ORDER M' (USUALLY $M \ll M'$)

Filter	Mult.	Div.	Add.
Laguerre	$17M + 2$	$14M$	$14M + 1$
FIR	$8M'$	$8M'$	$8M'$

we can also obtain the following time-update recursion for $\zeta_M^c(N)$:

$$\zeta_M^c(N) = a^2 \zeta_M^c(N-1) + \frac{|\check{c}_M(N)|^2}{\gamma_M(N)}$$

Fig. 4 illustrates the structure of the RLS-Laguerre Lattice algorithm, which is listed in Table II. Here again, we redefined the minimum cost variables in order to save addition operations, namely

$$\begin{aligned} \zeta_M^b(N) &\triangleq \mu + \zeta_M^b(N) \\ \zeta_M^f(N) &\triangleq \mu + \xi_M^f(N) \\ \zeta_M^{\bar{b}}(N) &\triangleq \mu + \xi_M^{\bar{b}}(N). \end{aligned}$$

Table III compares the computational cost of a Laguerre lattice filter of order M with a shift-structured (FIR) lattice filter of order M' with $\lambda = 1$ (recall that, in general, $M \ll M'$).

Comparing Fig. 4 with the conventional lattice structure of Fig. 2, we see that the new lattice filter is still fundamentally simple; the major modification is in the substitution of the delay blocks of Fig. 2 by a second lattice filter that runs in parallel. This, in effect, corresponds to replacing the delay blocks by simple time-variant lattice sections. We may also mention that the algorithm of Table II is based on propagating the *a posteriori* estimation errors. Alternative implementations that are based on *a priori* errors, or even on normalized errors, can be derived and will be pursued elsewhere [22]. This is in addition to array forms. Note also that in the listing of Table II, we employed time-updates for the variables $\{\zeta_M^f(N), \zeta_M^b(N), \zeta_M^{\bar{b}}(N)\}$; order-update relations are also possible and can be used.

V. SIMULATIONS

In order to illustrate the advantages of using a Laguerre-based adaptive lattice structure, we compare the performance of a sixth-order Laguerre lattice filter with a shift-structured RLS lattice implementation of order 500. For this purpose, we consider the same IIR system used in [18], *viz*

$$G(z) = \frac{0.0017z^{-1}(1 + 0.673z^{-1})}{(1 - 0.368z^{-1})(1 - 0.819z^{-1})(1 - 0.995z^{-1})}$$

The input signal is a first-order AR process, and the SNR at the output is 50 dB. The Laguerre pole is located at $a = 0.978$, as in [18]. Fig. 5 shows the resulting learning curves that are obtained by averaging over 20 experiments. It is clear that the mean-square error of the Laguerre structure is considerably better during the training phase. As expected, the Laguerre network is

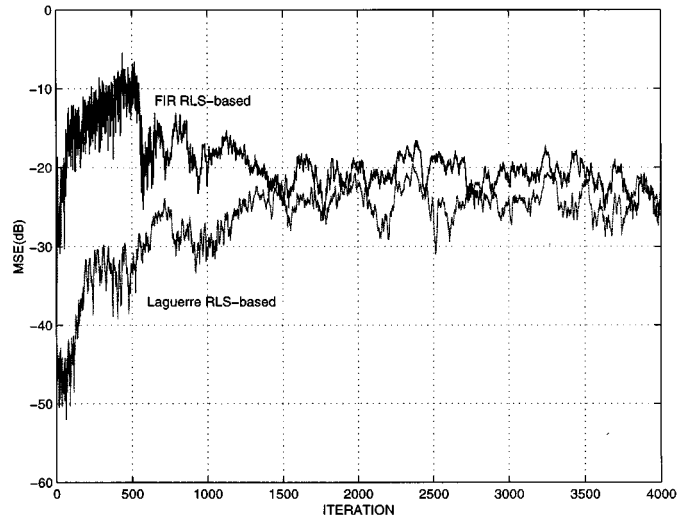


Fig. 5. Comparison of the learning curves of a 500-section RLS-FIR lattice with a six-section RLS-Laguerre lattice in an IIR modeling experiment.

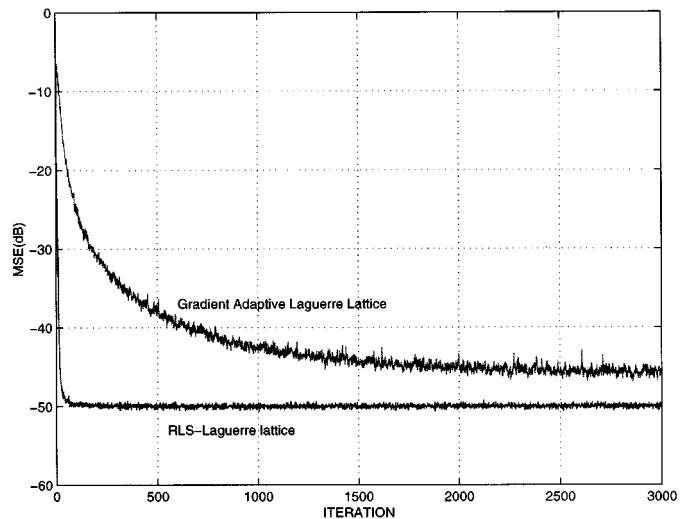


Fig. 6. Comparison of the learning curves of RLS-Laguerre lattice and gradient-Laguerre lattice.

better suited to modeling the IIR system, which is accomplished at significantly less computational burden.

We also compare the performance of the RLS-Laguerre algorithm with the corresponding gradient-Laguerre lattice algorithm proposed in [18]. We consider a system identification scenario where the unknown system to be identified is itself a Laguerre network of the same order $M = 6$ as the Laguerre adaptive filters. The input signal is simply white noise, and the Laguerre pole is fixed at $a = 0.5$. Fig. 6 shows the learning curves of both algorithms averaged over 1000 experiments. We see that the RLS-Laguerre lattice offers significant improvement in both convergence performance and misadjustment, as is expected for least-squares designs.

VI. CONCLUSIONS

We developed a framework for efficient order-recursive RLS adaptive filtering for input data that do not necessarily arise from tapped-delay lines. A special important case occurs in Laguerre-

based networks, where successive columns in the data matrix, while not shifted versions of each other, can still be related by a more general operation.

The approach of this paper can be extended to other filter networks, other than the Laguerre structure, especially when differences of the form $W - \Phi^*W\Phi$ have low rank. In addition, we can also develop normalized versions, array versions, and lattice schemes with feedback. These extensions will be published elsewhere [22].

REFERENCES

- [1] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive least-squares ladder estimation algorithms," *IEEE Trans. Circuits, Syst.*, no. 6, pp. 467–481, June 1981.
- [2] H. Lev-Ari, T. Kailath, and J. Cioffi, "Least-squares adaptive lattice and transversal filters: A unified geometrical theory," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 222–236, Mar. 1984.
- [3] B. Friedlander, "Lattice filters for adaptive processing," *Proc. IEEE*, vol. 70, pp. 829–867, Aug. 1982.
- [4] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [5] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias, *Advanced Digital Signal Processing*. New York: Macmillan, 1992.
- [6] P. Strobach, *Linear Prediction Theory*. Berlin/Heidelberg, Germany: Springer-Verlag, 1990.
- [7] A. H. Sayed and T. Kailath, "Extended Chandrasekhar recursions," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 619–623, Mar. 1994.
- [8] —, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Mag.*, vol. 11, pp. 18–60, July 1994.
- [9] G. Carayannis, D. Manolakis, and N. Kalouptsidis, "A fast sequential algorithm for least squares filtering and prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1394–1402, Dec. 1983.
- [10] J. Cioffi and T. Kailath, "Fast recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 304–337, Apr. 1984.
- [11] R. Merched and A. H. Sayed, "Fast RLS Laguerre adaptive filtering," in *Proc. Allerton Conf.*, Allerton, IL, Sept. 1999.
- [12] A. C. den Brinker, "Laguerre-domain adaptive filters," *IEEE Trans. Signal Processing*, vol. 42, pp. 953–956, Apr. 1994.
- [13] J. J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Mag.*, vol. 6, pp. 4–21, Apr. 1989.
- [14] J. C. Principe, B. de Vries, and P. G. de Oliveira, "The Gamma filter—A new class of adaptive IIR filters with restricted feedback," *IEEE Trans. Signal Processing*, vol. 41, pp. 649–656, Feb. 1993.
- [15] P. A. Regalia, *Adaptive IIR Filtering in Signal Processing and Control*. New York: Marcel Dekker, 1995.
- [16] P. M. Makila, "Approximation of stable systems by Laguerre filters," *Automatica*, vol. 26, pp. 333–345, Feb. 1990.
- [17] B. Wahlberg, "System identification using Laguerre models," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 551–562, May 1991.

- [18] Z. Fejzo and H. Lev-Ari, "Adaptive Laguerre-lattice filters," *IEEE Trans. Signal Processing*, vol. 45, pp. 3006–3016, Dec. 1997.
- [19] J. W. Davidson and D. D. Falconer, "Reduced complexity echo cancellation using orthonormal functions," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 20–28, Jan. 1991.
- [20] A. H. Sayed, "Adaptive filtering," unpublished lecture notes, Dept. Elect. Eng., Univ. Calif., Los Angeles, 1997.
- [21] T. O. Silva, "On the determination of the optimal pole position of Laguerre filters," *IEEE Trans. Signal Processing*, vol. 43, pp. 2079–2087, Sept. 1995.
- [22] R. Merched and A. H. Sayed, "RLS-LaGuerre adaptive filtering: Error-feedback, normalized, and array-based algorithms," submitted for publication.



Ricardo Merched (S'97) was born in Rio de Janeiro, Brazil. He received the B.S. and M.S. degrees in electrical engineering from Universidade Federal do Rio de Janeiro in 1995 and 1997, respectively. Since September 1997, he has been pursuing the Ph.D. degree with the Electrical Engineering Department, University of California, Los Angeles, working on different aspects of adaptive filter design, especially multirate adaptive filters and fast RLS Laguerre adaptive filters.



Ali H. Sayed (SM'99) received the Ph.D. degree in electrical engineering in 1992 from Stanford University, Stanford, CA.

He is Associate Professor of electrical engineering at the University of California, Los Angeles (UCLA). He has over 135 journal and conference publications, is co-author of the research monograph *Indefinite Quadratic Estimation and Control* (Philadelphia, PA: SIAM, 1999) and of the graduate-level textbook *Linear Estimation* (Englewood Cliffs, NJ: Prentice-Hall, 2000). He is also co-editor of the volume *Fast Reliable Algorithms for Matrices with Structure* (Philadelphia, PA: SIAM, 1999). He is a member of the editorial boards of the *SIAM Journal on Matrix Analysis and Its Applications* and the *International Journal of Adaptive Control and Signal Processing* and is Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He has served as Co-Editor of special issues of the journal *Linear Algebra and Its Applications*. He has contributed several articles to engineering and mathematical encyclopedias and handbooks. His research interests span several areas including adaptive and statistical signal processing, filtering and estimation theories, equalization techniques for communications, interplays between signal processing and control methodologies, and fast algorithms for large-scale problems.

Dr. Sayed has served on the program committees of several international meetings. He is also a recipient of the 1996 IEEE Donald G. Fink Award.