

On Extended RLS Lattice Adaptive Variants: Error-Feedback, Normalized, and Array-Based Recursions

Ricardo Merched

Signal Processing Laboratory (LPS), Department of Electronics and Computer Engineering, Federal University of Rio de Janeiro, P.O. Box 68504, Rio de Janeiro, RJ 21945-970, Brazil
 Email: merched@lps.ufrj.br

Received 12 May 2004; Revised 10 November 2004; Recommended for Publication by Hideaki Sakai

Error-feedback, normalized, and array-based recursions represent equivalent RLS lattice adaptive filters which are known to offer better numerical properties under finite-precision implementations. This is the case when the underlying data structure arises from a tapped-delay-line model for the input signal. On the other hand, in the context of a more general orthonormality-based input model, these variants have not yet been derived and their behavior under finite precision is unknown. This paper develops several lattice structures for the exponentially weighted RLS problem under orthonormality-based data structures, including error-feedback, normalized, and array-based forms. As a result, besides nonminimality of the new recursions, they present unstable modes as well as hyperbolic rotations, so that the well-known good numerical properties observed in the case of FIR models no longer exist. We verify via simulations that, compared to the standard extended lattice equations, these variants do not improve the robustness to quantization, unlike what is normally expected for FIR models.

Keywords and phrases: RLS algorithm, orthonormal model, lattice, regularized least squares.

1. INTRODUCTION

In a recent paper [1], a new framework for exploiting data structure in recursive-least-squares (RLS) problems has been introduced. As a result, we have shown how to derive RLS lattice recursions for more general orthonormal networks other than tapped-delay-line implementations [2]. As is well known, the original fast RLS algorithms are obtained by exploiting the shift structure property of the successive rows of the corresponding input data matrix to the adaptive algorithm. That is, consider two successive regression (row) vectors $\{u_{M,N}, u_{M,N+1}\}$, of order M , say,

$$\begin{aligned} u_{M,N} &= [u_0(N) \quad u_1(N) \quad \cdots \quad u_{M-1}(N)] \\ &= [u_{M-1,N} \quad u_{M-1}(N)], \\ u_{M,N+1} &= [u_0(N+1) \quad u_1(N+1) \quad \cdots \quad u_{M-1}(N+1)] \\ &= [u_0(N+1) \quad \tilde{u}_{M-1,N+1}]. \end{aligned} \quad (1)$$

By recognizing that, in tapped-delay-line models we have

$$\tilde{u}_{M-1,N+1} = u_{M-1,N}. \quad (2)$$

One can exploit this relation to obtain the LS solution in a fast manner. The key for extending this concept to more

general structures in [1, 3] was to show that, although the above equality no longer holds for general orthonormal models, it is still possible to relate the entries of $\{u_{M,N}, u_{M,N+1}\}$ as

$$\tilde{u}_{M-1,N+1} = u_{M,N} \Phi_M, \quad (3)$$

where Φ_M is an $M \times (M - 1)$ structured matrix induced by the underlying orthonormal model. Figure 1 illustrates such structure for which the RLS lattice algorithm of [1] was derived. They constitute what we will refer to in this paper as the *a-posteriori*-based lattice algorithm, since all these recursions are based on a *posteriori* estimation errors. Now, it is a well-understood fact that several other equivalent lattice structures exist for RLS filters that result from tapped-delay-line models. These alternative implementations are known in the literature as *error-feedback*, *array-based* (also referred to as *QRD lattice*), and *normalized lattice* algorithms (see, e.g., [4, 5, 6, 7, 8]). In [9], all such variants were further extended to the special case of Laguerre-based filters, as we have explained in [1]. Although all these forms are theoretically equivalent, they tend to exhibit different performances when considered under finite-precision effects.

In this paper, we will derive all such equivalent lattice implementations for input data models based on the structure

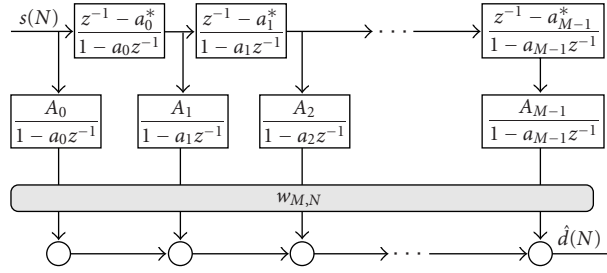


FIGURE 1: Transversal orthonormal structure for adaptive filtering.

of Figure 1. The use of orthonormal bases can provide several advantages. For example, in some situations, long FIR models can be replaced by shorter compact all-pass-like models as Laguerre filters (see, e.g., [10, 11]). From the adaptive filtering point of view, this can represent large savings in computational complexity. The conventional IIR adaptive methods [12, 13] present serious problems of stability, local minima, and slow convergence, and in this sense the use of orthogonal bases offers a stable and global solution, due to their fixed poles location. Moreover, orthonormality guarantees good numerical conditioning for the underlying estimation problem, in contrast to other equivalent system descriptions (such as the fixed-denominator model and the partial-fraction representation—see further [2]). The most important application of such structured RLS problems is in the field of line echo cancelation corresponding to long channels, whereby FIR models can be replaced by short orthonormal IIR models. Other applications include channel-estimate-based equalization schemes, where the feedforward linear equalizer can be similarly replaced by an orthonormal IIR structure.

After obtaining the new algorithms we will verify their performance through computer simulations under finite-precision arithmetic. As a result, the new forms turn out to exhibit an unstable behavior. Besides nonminimality of their corresponding algorithm states, they present unstable modes or hyperbolic rotations in their recursions, unlike the corresponding fast variants for FIR models (the latter, in contrast, is free from hyperbolic rotations and unstable modes, and present better numerical properties, despite nonminimality). As a consequence, the new variants do not show improvement in robustness to the quantization effect, compared to the standard RLS lattice recursions of [1], which remains the only reliable extended lattice structure. This discussion on the numerical effects is provided in Section 9.

However, before starting our presentation we call the attention of the reader for an important point. Our main goal in this paper is the development of the equivalent RLS recursions that are normal extensions of the FIR case, and to present some preliminary comparisons based on computer simulations. A complete analytical error analysis for each of

these algorithms is not a simple task and is beyond the scope of this paper. Nevertheless, the algorithms derivation is by itself a starting point for further development and improvements on such variants, which is a subject for future research. Moreover, we will provide a brief review and discussion on the minimality and backward consistency properties in order to explain (up to a certain extent) the stability of these variants from the point of view of error propagation. This will be pursued in Section 9, while commenting on the sources of numerical errors in each case.

Notation. In this paper, $A \oplus B$ is the same as $\text{diag}\{A, B\}$. We also denote $*$ as the conjugate and transpose of a vector. Since we will be dealing with order-recursive variables, we will write, for example, $H_{M,N}$, the order- M data matrix up to time N . The same goes for $u_{M,N}$, $e_M(N)$, and so on.

2. A MODIFIED RLS ALGORITHM

We first provide a brief review of the regularized least-squares problem, but with a slight modification in the definitions of the desired vector, denoted by y_N , and the weighting matrix \mathcal{W}_N . Thus given a column vector $y_N \in \mathbb{C}^{N+1}$, and a data matrix $H_N \in \mathbb{C}^{(N+1) \times M}$, the exponentially weighted least-squares problem seeks the column vector $w \in \mathbb{C}^M$ that solves

$$\min_{w_M} \left[\lambda^{N+1} w_M^* \Pi_M^{-1} w_M + \|d_N - \mathcal{H}_{M,N} w_M\|_{\mathcal{W}_N}^2 \right], \quad (4)$$

where Π_M is a positive regularization matrix, $\mathcal{W}_N = (\lambda^N \oplus \lambda^{N-1} \oplus \dots \oplus \lambda \oplus t)$ is a weighting matrix defined in terms of a forgetting factor λ satisfying $0 \ll \lambda < 1$, and t is an arbitrary scaling factor. The symbol $*$ denotes complex conjugate transposition. Moreover, we define d_N as a growing length vector whose entries are assumed to change according to the following rule:

$$d_N = \begin{bmatrix} \theta d_{N-1} \\ d(N) \end{bmatrix} \quad (5)$$

for some scalar θ .¹ The individual rows of \mathcal{H}_N are denoted by $\{u_i\}$:

$$\mathcal{H}_{M,N} = \begin{bmatrix} u_{M,0} \\ u_{M,1} \\ \vdots \\ u_{M,N} \end{bmatrix}. \quad (6)$$

Note that the regularized problem in (4) can be conveniently written as

$$\min_{w_M} \left\| \begin{bmatrix} 0_L \\ d_N \end{bmatrix} - \begin{bmatrix} \mathcal{A}_{M,L} \\ \mathcal{H}_{M,N} \end{bmatrix} w_M \right\|_{W_N}^2, \quad (7)$$

where $W_N = (\lambda^{N+L} \oplus \lambda^{N+L-1} \oplus \dots \oplus t)$, and where we have factored Π_M^{-1} as

$$\Pi_M^{-1} = \mathcal{A}_{M,L}^* \mathcal{W}_L \mathcal{A}_{M,L} \quad (8)$$

for some matrix $\mathcal{A}_{M,L}$. This assumes that the incoming data has started at some point in the past, depending on the number of rows L of $\mathcal{A}_{M,L}$ (see [1]). Hence, defining the extended quantities

$$H_{M,N} \triangleq \begin{bmatrix} \mathcal{A}_{M,L} \\ \mathcal{H}_{M,N} \end{bmatrix} = \begin{bmatrix} x_{0,-1} & x_{1,-1} & \dots & x_{M-1,-1} \\ h_{0,N} & h_{1,N} & \dots & h_{M-1,N} \end{bmatrix} \quad (9)$$

$$\triangleq [x_{0,N} \ x_{1,N} \ \dots \ x_{M-1,N}],$$

where $x_{i,-1}$ represents a column of $\mathcal{A}_{M,L}$ and $h_{i,N}$ denotes a column of $\mathcal{H}_{M,N}$, as well as

$$y_N = \begin{bmatrix} 0_L \\ d_N \end{bmatrix}, \quad (10)$$

we can express (4) as a pure least-squares problem:

$$\min_{w_M} \|y_N - H_{M,N} w_M\|_{W_N}^2. \quad (11)$$

Therefore, the optimal solution of (11), denoted by $w_{M,N}$, is given by

$$w_{M,N} \triangleq P_{M,N} H_{M,N}^* W_N y_N, \quad (12)$$

where

$$P_{M,N} = (H_{M,N}^* W_N H_{M,N})^{-1}. \quad (13)$$

We denote the projection of y_N onto the range space of H_N by $y_{M,N} = H_{M,N} w_{M,N}$. The corresponding *a posteriori* estimation error vector is given by $e_N = y_N - H_{M,N} w_{M,N}$.

Now let $w_{M,N-1}$ be the solution to a similar LS problem with the variables $\{y_N, H_{M,N}, W_N, \lambda^{N+1}\}$ in (4) replaced by $\{y_{N-1}, H_{M,N-1}, W_{N-1}, \lambda^N\}$. That is,

$$w_{M,N-1} = (H_{M,N-1}^* W_{N-1} H_{M,N-1})^{-1} H_{M,N-1}^* W_{N-1} y_{N-1}. \quad (14)$$

Using (5) and the fact that

$$H_{M,N} = \begin{bmatrix} H_{M,N-1} \\ u_{M,N} \end{bmatrix} \quad (15)$$

in addition to the matrix inversion formula, it is straightforward to verify that the following (modified) RLS recursions hold:

$$\begin{aligned} \gamma_M^{-1}(N) &= 1 + t\lambda^{-1} u_{M,N} P_{M,N-1} u_{M,N}^*, \\ g_{M,N} &= \lambda^{-1} P_{M,N-1} u_{M,N}^* \gamma_M(N), \\ w_{M,N} &= \theta w_{M,N-1} + t g_{M,N} \epsilon_M(N), \\ \epsilon_M(N) &= d(N) - \theta u_{M,N} w_{M,N-1}, \\ P_{M,N} &= \lambda^{-1} P_{M,N-1} - g_{M,N} \gamma_M^{-1}(N) g_{M,N}^*, \end{aligned} \quad (16)$$

with $w_{M,-1} = 0_M$ and $P_{M,-1} = \Pi_M$. These recursions tell us how to update the weight estimate $w_{M,N}$ in time. The well-known exponentially weighted RLS algorithm corresponds to the special choice $\theta = t = 1$. The introduction of the scalars $\{\theta, t\}$ allows for a level of generality that is convenient for our purposes in the coming sections.

3. STANDARD LATTICE RECURSIONS

Algorithm 1 shows the standard extended lattice recursions that solves the RLS problem when the underlying input regression vectors arise from the orthonormal network of Figure 1. The matrix Π_M as well as all the initialization variables are obtained according to an offline procedure as described in [1]. The main step in this initialization procedure is the computation of Π_M , which remains unchanged for the new recursions we will present in the next sections. The reader should refer to [1] for the details of its computation. Figure 2 illustrates the structure of the m th section of this extended lattice algorithm.

4. ERROR-FEEDBACK LATTICE FILTERS

Observe that all the reflection coefficients defined for the a-posteriori-based lattice algorithm are computed as a ratio in which the numerator and denominator are updated via separate recursions. An error-feedback form is one that replaces the individual recursions for the numerator and denominator quantities by equivalent recursions for the reflection coefficients themselves. In principle, one could derive the recursions algebraically as follows. Consider for instance

$$\kappa_M(N) = \frac{\rho_M(N)}{\zeta_M^b(N)}. \quad (17)$$

¹The reason for the introduction of the scalars $\{\theta, t\}$ will be understood very soon. The classical recursive least-squares (RLS) problem corresponds to the special choice $\theta = t = 1$.

<i>Initialization</i>
<p>For $m = 0$ to M, set</p> $\zeta_m^f(-1) = \mu \text{ (small positive number)}$ $\delta_m(-1) = \rho_m(-1) = v_m(-1) = b_m(-1) = 0$ $\zeta_m^b(-1) = \pi_m - c_m^* \Pi_m c_m$ $\check{\zeta}_m^b(-1) = \check{\pi}_{m+1} - \check{c}_m^* \check{\Pi}_m \check{c}_m$ $\sigma_m = \lambda \zeta_m^b(-1) / \zeta_m^f(-1)$ $\chi_m(-1) = a_m \phi_m^* \Pi_m c_m + A_m$ $\zeta_m^b(-1) = \zeta_{m+1}^b(-1)$
<p>For $N \geq 0$, repeat</p> $\gamma_0(N) = \bar{\gamma}_0(N) = 1, \quad f_0(N) = b_0(N) = s(N)$ $v_0(N) = 0, \quad e_0(N) = d(N)$ <p>For $m = 0$ to $M - 1$, repeat</p> $\check{\zeta}_m^b(N) = \sigma_m \bar{\gamma}_m(N) \zeta_m^f(N - 1)$ $\kappa_m^b(N) = \check{\zeta}_m^b(N) \chi_{m+1}(N - 1)$ $\bar{b}_m(N) = a_{m+1} b_{m+1}(N - 1) + \kappa_m^b(N) v_{m+1}(N - 1)$ $\zeta_m^f(N) = \lambda \zeta_m^f(N - 1) + f_m(N) ^2 / \bar{\gamma}_m(N)$ $\zeta_m^b(N) = \lambda \zeta_m^b(N - 1) + b_m(N) ^2 / \gamma_m(N)$ $\check{\zeta}_m^b(N) = \lambda \check{\zeta}_m^b(N - 1) + \bar{b}_m(N) ^2 / \bar{\gamma}_m(N)$ $\chi_m(N) = \chi_m(N - 1) + a_m v_m^*(N) \beta_m(N)$ $\delta_m(N) = \lambda \delta_m(N - 1) + f_m^*(N) \bar{b}_m(N) / \bar{\gamma}_m(N)$ $\rho_m(N) = \lambda \rho_m(N - 1) + e_m^*(N) b_m(N) / \gamma_m(N)$ $\gamma_{m+1}(N) = \gamma_m(N) - b_m(N) ^2 / \zeta_m^b(N)$ $\bar{\gamma}_{m+1}(N) = \bar{\gamma}_m(N) - \bar{b}_m(N) ^2 / \check{\zeta}_m^b(N)$ $\kappa_m^v(N) = \chi_m^*(N) / \zeta_m^b(N), \quad \kappa_m(N) = \rho_m(N) / \zeta_m^b(N)$ $\kappa_m^f(N) = \delta_m(N) / \zeta_m^f(N), \quad \kappa_m^f(N) = \delta_m^*(N) / \zeta_m^b(N)$ $v_{m+1}(N) = -a_m^* v_m(N) + \kappa_m^v(N) b_m(N)$ $e_{m+1}(N) = e_m(N) - \kappa_m(N) b_m(N)$ $b_{m+1}(N) = \bar{b}_m(N) - \kappa_m^b(N) f_m(N)$ $f_{m+1}(N) = f_m(N) - \kappa_m^f(N) \bar{b}_m(N)$
<i>Alternative recursions</i>
$\zeta_{m+1}^f(N) = \zeta_m^f(N) - \delta_m(N) ^2 / \zeta_m^b(N)$ $\zeta_{m+1}^b(N) = \zeta_m^b(N) - \delta_m(N) ^2 / \zeta_m^f(N)$ $\zeta_m^v(N) = \lambda^{-1} \zeta_m^v(N - 1) - v_m(N) ^2 / \gamma_m(N)$ $\check{\zeta}_m^b(N) = a_{m+1} ^2 \zeta_{m+1}^b(N - 1) + \zeta_m^b(N) \chi_{m+1}(N - 1) ^2$ $\bar{\gamma}_m(N) = \gamma_{m+1}(N - 1) + \zeta_m^b(N) v_{m+1}(N - 1) ^2$

ALGORITHM 1: Standard extended RLS lattice recursions.

From the listing of the *a-posteriori*-based lattice filter of Algorithm 1, substituting the recursions for $\rho_m(N)$ and $\zeta_m^b(N)$ into the expression for $\kappa_m(N)$ leads to

$$\kappa_M(N) = \frac{\rho_M(N - 1) + e_M^*(N) b_M(N) / \gamma_M(N)}{\zeta_M^b(N - 1) + |b_M(N)|^2 / \gamma_M(N)} \quad (18)$$

and some algebra will result in a relation between $\kappa_M(N)$ and $\kappa_M(N - 1)$.

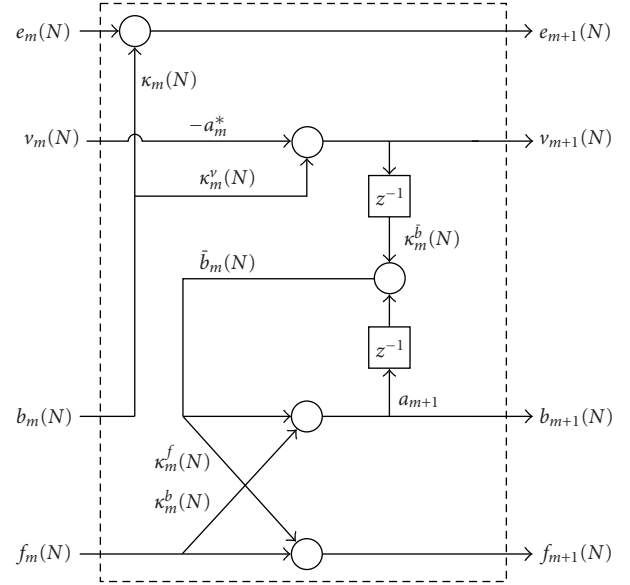


FIGURE 2: A lattice section.

We will not pursue this algebraic procedure here. Instead, we will follow the arguments used in [9] which highlights the interpretation of the reflection coefficients in terms of a least-squares problem. This will allow us to invoke the recursions we have already established for the modified RLS problem of Section 2 and to arrive at the recursions for the reflection coefficients almost by inspection.

4.1. A priori estimation errors

One form of error-feedback algorithm is the one based on *a priori*, as opposed to a *posteriori*, estimation errors. They are defined as

$$\begin{aligned} \beta_{M+1,N} &= x_{M+1,N} - H_{M+1,N} w_{M+1,N-1}^b, \\ \bar{\beta}_{M,N} &= x_{M+1,N} - \bar{H}_{M,N} w_{M,N-1}^b, \\ \alpha_{M+1,N} &= x_{0,N} - \bar{H}_{M,N} w_{M,N-1}^f, \\ \epsilon_{M,N} &= y_N - H_{M,N} w_{M,N-1}, \end{aligned} \quad (19)$$

where now the *a posteriori* weight vector $w_{M,N}^f$, for example, is replaced by $w_{M,N-1}^f$. That is, these recursions are similar to the ones used for the *a posteriori* errors $\{e_{M,N}, \bar{b}_{M,N}, b_{M+1,N}, f_{M+1,N}\}$, with the only difference lying in the use of prior weight vector estimates.

Following the same arguments as in Section III of [1], it can be verified that the last entries of these errors satisfy the following order-update relations in terms of the same reflection coefficients $\{\kappa_M(N), \kappa_M^f(N), \kappa_M^b(N)\}$:

$$\begin{aligned} \epsilon_{M+1}(N) &= \epsilon_M(N) - \kappa_M(N - 1) \beta_M(N), \\ \beta_{M+1}(N) &= \bar{\beta}_M(N) - \kappa_M^b(N - 1) \alpha_M(N), \\ \alpha_{M+1}(N) &= \alpha_M(N) - \kappa_M^f(N - 1) \bar{\beta}_M(N), \end{aligned} \quad (20)$$

where $\{\kappa_M^f(N), \kappa_M^b(N), \kappa_M(N)\}$ can be updated as

$$\begin{aligned}\kappa_M^f(N) &= \kappa_M^f(N-1) + \frac{\bar{\beta}_M^*(N)\bar{\gamma}_M(N)}{\zeta_M^b(N)}\alpha_{M+1}(N), \\ \kappa_M^b(N) &= \kappa_M^b(N-1) + \frac{\alpha_M^*(N)\bar{\gamma}_M(N)}{\zeta_M^f(N)}\beta_{M+1}(N), \\ \kappa_M(N) &= \kappa_M(N-1) + \frac{\beta_M^*(N)\gamma_M(N)}{\zeta_M^b(N)}\epsilon_{M+1}(N).\end{aligned}\quad (21)$$

The above recursions are well known and they are obtained regardless of data structure.

Now, recall that the a-posteriori-based algorithm still requires the recursions for $\{\bar{b}_M(N), \nu_M(N)\}$, where $\nu_M(N)$ is referred to as the *a posteriori rescue variable*. As we will see in the upcoming sections, similar arguments will also lead to the quantities $\{\bar{\beta}_M(N), \nu_M(N)\}$, where $\nu_M(N)$ will be similarly defined as the *a priori rescue variable* corresponding to $\nu_M(N)$. These in turn will allow us to obtain recursions for their corresponding reflection coefficients $\{k_M^b(N), k_M^v(N)\}$. Moreover, we will verify that $\nu_M(N)$ is the actual rescue quantity used in the fixed-order fast transversal algorithm, and which is based on a priori estimation errors.

4.2. Exploiting data structure

The procedure to find a recursion for $\beta_{M,N}$ follows similarly to the one for the a posteriori error $b_{M,N}$. Thus, beginning from its definition

$$\begin{aligned}\bar{\beta}_{M,N} &= x_{M+1,N} - \bar{H}_{M,N}\bar{P}_{M,N-1}\bar{H}_{M,N-1}^*W_{N-1}x_{M+1,N-1} \\ &= x_{M+1,N} - \begin{bmatrix} 0 \\ H_{M+1,N-1} \end{bmatrix}\Phi_{M+1}\bar{P}_{M,N-1} \\ &\quad \times \Phi_{M+1}^*\begin{bmatrix} 0 & H_{M+1,N-2}^* \end{bmatrix}W_{N-1}x_{M+1,N-1},\end{aligned}\quad (22)$$

where Φ is the matrix that relates $\{H_{M+1,N-1}, \bar{H}_{M,N}\}$, and using the following relations into (22) (see [1]):

$$\begin{aligned}\Phi_{M+1}\bar{P}_{M,N-1}\Phi_{M+1}^* &= P_{M+1,N-2} - \zeta_M^b(N-1)P_{M+1,N-2}\phi_{M+1}\phi_{M+1}^*P_{M+1,N-2}, \\ x_{M+1,N-1} &= a_{M+1}\begin{bmatrix} 0 \\ x_{M+1,N-2} \end{bmatrix} \\ &\quad + \frac{A_{M+1}}{A_M}\left(\begin{bmatrix} 0 \\ x_{M,N-2} \end{bmatrix} - a_M^*x_{M,N-1}\right),\end{aligned}\quad (23)$$

we obtain, after some algebra,

$$\begin{aligned}\bar{\beta}_M(N) &= a_{M+1}\bar{\beta}_{M+1}(N-1) \\ &\quad + \zeta_M^b(N-1)\chi_{M+1}(N-2)\lambda k_{M+1,N-1}^*\phi_{M+1},\end{aligned}\quad (24)$$

where

$$k_{M,N} = g_{M,N}\gamma_M^{-1}(N)\quad (25)$$

is the normalized gain vector, defined by the corresponding fast fixed-order recursions. Thus, defining the a priori rescue variable

$$\nu_M(N) \triangleq k_{M+1,N-1}^*\phi_{M+1},\quad (26)$$

we have

$$\begin{aligned}\bar{\beta}_M(N) &= a_{M+1}\bar{\beta}_{M+1}(N-1) \\ &\quad + \lambda k_M^b(N-1)\nu_{M+1}(N-1).\end{aligned}\quad (27)$$

In order to obtain a recursion for $\nu_M(N)$, consider the order-update recursion for $k_{M,N}$, that is,

$$k_{M+1,N} = \begin{bmatrix} k_{M,N} \\ 0 \end{bmatrix} + \frac{\beta_M^*(N)}{\lambda\zeta_M^b(N-1)}\begin{bmatrix} -w_{M,N-1}^b \\ 1 \end{bmatrix}.\quad (28)$$

Taking the complex transposition of (28) and multiplying it from the left by ϕ_{M+1} we get

$$\nu_{M+1}(N) = -a_M^*\nu_M(N) + \lambda^{-1}\kappa_M^v(N-1)\beta_M(N).\quad (29)$$

Of course, an equivalent recursion for $\chi_M(N)$ can be obtained, considering the time update for $w_{M,N}^b$, which can be written as

$$\begin{bmatrix} -w_{M,N}^b \\ 1 \end{bmatrix} = \begin{bmatrix} -w_{M,N-1}^b \\ 1 \end{bmatrix} - \begin{bmatrix} k_{M,N} \\ 0 \end{bmatrix}b_M(N).\quad (30)$$

Hence, multiplying (30) from the left by ϕ_{M+1}^* we get

$$\chi_M(N) = \chi_M(N-1) + a_M\nu_M^*(N)b_M(N).\quad (31)$$

Now, it only remains to find recursions for the reflection coefficients $\{k_M^b(N), k_M^v(N)\}$.

4.3. Time updates for $\{k_M^b(N), k_M^v(N)\}$

We now obtain time relations for the reflection coefficients by exploiting the fact that these coefficients can be regarded as least-squares solutions of order one [9, 14].

We begin with the reflection coefficient

$$k_M^b(N) = \zeta_M^b(N)\chi_{M+1}(N-1) = \frac{\chi_{M+1}(N-1)}{\zeta_{M+1}^v(N-1)},\quad (32)$$

where, from (31) and Section 5.1 of [1], the numerator and denominator quantities satisfy

$$\begin{aligned}\chi_M(N) &= \chi_M(N-1) + a_M\nu_M^*(N)b_M(N), \\ \zeta_M^v(N) &= \lambda^{-1}\zeta_M^v(N-1) - \frac{|\nu_M(N)|^2}{\gamma_M(N)}.\end{aligned}\quad (33)$$

Now define the *angle normalized* errors

$$\begin{aligned} b'_M(N) &\triangleq \frac{b_M(N)}{\gamma_M^{1/2}(N)} = \beta_M(N)\gamma_M^{1/2}(N), \\ v'_M(N) &\triangleq \frac{v_M(N)}{\gamma_M^{1/2}(N)} = \nu_M(N)\gamma_M^{1/2}(N) \end{aligned} \quad (34)$$

in terms of the square root of the conversion factor $\gamma_M(N)$. It then follows from the above time updates for $\chi_M(N)$ and $\zeta_M^v(N)$ that $\{\chi_M(N), \zeta_M^v(N)\}$ can be recognized as the inner products

$$\begin{aligned} \chi_M(N) &= a_M v'^*_{M,N} b'_{M,N}, \\ \zeta_M^v(N) &= v'^*_{M,N} W_N^{-1} v'_{M,N}, \end{aligned} \quad (35)$$

which are written in terms of the following vectors of angle normalized prediction errors:

$$b'_{M,N} \triangleq \begin{bmatrix} b'_M(-L) \\ b'_M(-L+1) \\ \vdots \\ b'_{M+1}(N) \end{bmatrix}, \quad v'_{M,N} \triangleq \begin{bmatrix} v'_M(-L) \\ v'_M(-L+1) \\ \vdots \\ v'_M(N) \end{bmatrix}. \quad (36)$$

In this way, the defining relation (32) for $\kappa_M^{\bar{b}}(N)$ can be written as

$$\begin{aligned} \kappa_M^{\bar{b}}(N) &= \left(v'^*_{M+1,N-1} W_N^{-1} v'_{M+1,N-1} \right)^{-1} \\ &\quad \times v'^*_{M+1,N-1} W_N^{-1} \left(a_{M+1} W_N b'_{M,N-1} \right) \end{aligned} \quad (37)$$

which shows that $\kappa_M^{\bar{b}}(N)$ can be interpreted as the solution of a first-order weighted least-squares problem, namely that of projecting the vector $(a_{M+1} W_N b'_{M,N})$ onto the vector $v'_{M+1,N-1}$. This simple observation shows that $\kappa_M^{\bar{b}}(N)$ can be readily time updated by invoking the modified RLS recursion introduced in Section 2. That is, by making the identification $\theta \rightarrow \lambda$, and $t \rightarrow -1$, we have

$$\begin{aligned} \kappa_M^{\bar{b}}(N) &= \lambda \kappa_M^{\bar{b}}(N-1) \\ &\quad - \frac{v'^*_{M+1}(N-1)}{\zeta_M^v(N)} \left[-a_{M+1} b'_{M+1}(N-1) \right. \\ &\quad \quad \left. - \lambda v'_{M+1}(N-1) \kappa_M^{\bar{b}}(N-1) \right] \\ &= \lambda \kappa_M^{\bar{b}}(N-1) \\ &\quad + \frac{v'^*_{M+1}(N-1)}{\zeta_M^v(N)} \left[a_{M+1} \beta_{M+1}(N-1) \right. \\ &\quad \quad \left. + \lambda \gamma_{M+1}(N-1) \kappa_M^{\bar{b}}(N-1) \right] \\ &= \lambda \kappa_M^{\bar{b}}(N-1) + \zeta_M^{\bar{b}}(N) v'^*_{M+1}(N-1) \bar{\beta}_M(N). \end{aligned} \quad (38)$$

This last equation is obtained from the update for $\bar{\beta}_M(N)$

in (27). Similarly, the weight $\kappa_M^v(N)$ can be expressed as

$$\kappa_M^v(N) = \left(b'^*_{M,N} W_N b'_{M,N} \right)^{-1} b'^*_{M,N} W_N \left(a_M^* W_N^{-1} v'_{M,N} \right) \quad (39)$$

and therefore, making the identification $\theta = \lambda^{-1}$, and $t \rightarrow 1$, we can justify the following time update:

$$\begin{aligned} \kappa_M^v(N) &= \lambda^{-1} \kappa_M^v(N-1) \\ &\quad + \frac{b'^*_{M}(N)}{\zeta_M^b(N)} \left[a_M^* v'_M(N) - \lambda^{-1} b'_M(N) \kappa_M^v(N-1) \right] \\ &= \lambda^{-1} \kappa_M^v(N-1) - \frac{b_M^*(N)}{\zeta_M^b(N)} \nu_{M+1}(N). \end{aligned} \quad (40)$$

A similar approach will also lead to the time updates of $\{\kappa_M^f(N), \kappa_M^b(N), \kappa_M(N)\}$ defined previously. Algorithm 2 shows the a-priori-based lattice recursions with error feedback.²

5. A-POSTERIORI-BASED REFLECTION COEFFICIENT RECURSIONS

Alternative recursions for the reflection coefficients $\{\kappa_M^v(N), \kappa_M^f(N), \kappa_M^b(N), \kappa_M(N)\}$ that are based on a posteriori errors can also be obtained. The resulting reflection coefficients updates possess the advantage of avoiding the multiplicative factor λ^{-1} in the corresponding error-feedback recursions, which represent a potential source of instability of the algorithm.

Thus consider for example the first equality of (38). It can be written as

$$\begin{aligned} \kappa_M^{\bar{b}}(N) &= \left(1 + \frac{|v'_{M+1}(N-1)|^2}{\zeta_M^v(N)} \right) \lambda \kappa_M^{\bar{b}}(N-1) \\ &\quad + \frac{a_{M+1} v'^*_{M+1}(N-1) b_{M+1}(N-1)}{\gamma_{M+1}(N-1) \zeta_M^v(N)}. \end{aligned} \quad (41)$$

Recalling that $\bar{\gamma}_M(N)$ has the update

$$\bar{\gamma}_M(N) = \gamma_{M+1}(N-1) + \frac{|v_{M+1}(N-1)|^2}{\zeta_M^v(N)}, \quad (42)$$

we have that

$$\begin{aligned} \frac{\bar{\gamma}_M(N)}{\gamma_{M+1}(N-1)} &= \frac{\zeta_{M+1}^v(N-2)}{\lambda \zeta_{M+1}^v(N-1)} = \frac{\zeta_M^{\bar{b}}(N)}{\lambda \zeta_M^b(N-1)} \\ &= \left(1 + \frac{|v'_{M+1}(N-1)|^2}{\zeta_M^v(N)} \right) \end{aligned} \quad (43)$$

²Observe that the standard lattice filter obtained in [1] performs feedback of several estimation error quantities from a higher-order problem, for example, $b_{M+1}(N-1)$, into the computation of $b_M(N)$. The definition of *error feedback* in fast adaptive filters, however, has been referred to as the feedback of such estimation errors into the computation of the reflection coefficients themselves instead.

<i>Initialization</i>
<p>For $m = 0$ to M, set</p> <p>μ is a small positive number</p> <p>$\kappa_m(-1) = \kappa_m^b(-1) = \kappa_m^f(-1)$</p> <p>$\gamma_m(-1) = \beta_m(-1) = 0$</p> <p>$\zeta_m^f(-1) = \mu$</p> <p>$\zeta_m^b(-1) = \pi_m - c_m^* \Pi_m c_m$</p> <p>$\check{\zeta}_m^b(-1) = \check{\pi}_{m+1} - \check{c}_m^* \check{\Pi}_m \check{c}_m$</p> <p>$\sigma_m = \lambda \zeta_m^b(-1) / \zeta_m^f(-1)$</p> <p>$\chi_m(-1) = a_m \phi_m^* \Pi_m c_m + A_m$</p> <p>$\kappa_m^b(-1) = \zeta_m^b(-1) \chi_m(-1)$</p> <p>$\kappa_m^v(-1) = \chi_m^*(-1) / \zeta_m^b(-1)$</p> <p>$\check{\zeta}_m^b(-1) = \zeta_{m+1}^b(-1)$</p>
<p>For $N \geq 0$, repeat</p> <p>$\gamma_0(N) = \bar{\gamma}_0(N) = 1, \quad \alpha_0(N) = \beta_0(N) = s(N)$</p> <p>$\nu_0(N) = 0, \quad \epsilon_0(N) = d(N)$</p> <p>For $m = 0$ to $M - 1$, repeat</p> <p>$\check{\zeta}_m^b(N) = \sigma_m \bar{\gamma}_m(N) \zeta_m^f(N - 1)$</p> <p>$\bar{\beta}_m(N) = a_{m+1} \bar{\beta}_{m+1}(N - 1) + \lambda \kappa_M^b(N - 1) \gamma_{m+1}(N - 1)$</p> <p>$\kappa_M^b(N) = \lambda \kappa_M^b(N - 1) + \zeta_M^b(N) \nu_{M+1}^*(N - 1) \bar{\beta}_m(N)$</p> <p>$\zeta_m^f(N) = \lambda \zeta_m^f(N - 1) + \alpha_m(N) ^2 \bar{\gamma}_m(N)$</p> <p>$\check{\zeta}_m^b(N) = \lambda \zeta_m^b(N - 1) + \beta_m(N) ^2 \gamma_m(N)$</p> <p>$\zeta_m^b(N) = \lambda \zeta_m^b(N - 1) + \bar{\beta}_m(N) ^2 \bar{\gamma}_m(N)$</p> <p>$\gamma_{m+1}(N) = -a_m^* \gamma_m(N) + \lambda^{-1} \kappa_m^v(N - 1) \beta_m(N)$</p> <p>$\epsilon_{m+1}(N) = \epsilon_m(N) - \kappa_m(N - 1) \beta_m(N)$</p> <p>$\beta_{m+1}(N) = \bar{\beta}_m(N) - \kappa_m^b(N - 1) \alpha_m(N)$</p> <p>$\alpha_{m+1}(N) = \alpha_m(N) - \kappa_m^f(N - 1) \bar{\beta}_m(N)$</p> <p>$\kappa_M^v(N) = \lambda^{-1} \kappa_M^v(N - 1) - \frac{\beta_M^*(N) \gamma_M(N)}{\zeta_M^b(N)} \nu_{M+1}(N)$</p> <p>$\kappa_M^f(N) = \kappa_M^f(N - 1) + \frac{\beta_M^*(N) \bar{\gamma}_M(N)}{\zeta_M^b(N)} \alpha_{M+1}(N)$</p> <p>$\kappa_M^b(N) = \kappa_M^b(N - 1) + \frac{\alpha_M^*(N) \bar{\gamma}_M(N)}{\zeta_M^f(N)} \beta_{M+1}(N)$</p> <p>$\kappa_M(N) = \kappa_M(N - 1) + \frac{\beta_M^*(N) \gamma_M(N)}{\zeta_M^b(N)} \epsilon_{M+1}(N)$</p> <p>$\gamma_{m+1}(N) = \gamma_m(N) - \frac{ b_m(N) ^2}{\zeta_m^b(N)}$</p> <p>$\bar{\gamma}_{m+1}(N) = \bar{\gamma}_m(N) - \frac{ \bar{b}_m(N) ^2}{\zeta_m^b(N)}$</p>

ALGORITHM 2: The a-priori-based extended RLS lattice filter with error feedback.

so that we can write (41) as

$$\begin{aligned} \kappa_M^b(N) &= \frac{\lambda \bar{\gamma}_M(N)}{\gamma_{M+1}(N - 1)} \\ &\times \left(\kappa_M^b(N - 1) \right. \\ &\quad \left. + \frac{a_{M+1} \check{\zeta}_M^b(N - 1) \nu_{M+1}^*(N - 1) b_{M+1}(N - 1)}{\gamma_{M+1}(N - 1)} \right). \end{aligned} \quad (44)$$

In a similar fashion, we can obtain the following recursion for $\kappa_M^v(N)$ from (40):

$$\kappa_M^v(N) = \frac{\gamma_{M+1}(N)}{\lambda \gamma_M(N)} \left(\kappa_M^v(N - 1) + \frac{a_M^* b_M^*(N) \nu_M(N)}{\gamma_M(N) \zeta_M^b(N - 1)} \right), \quad (45)$$

where we have used the fact that

$$\frac{\gamma_{M+1}(N)}{\gamma_M(N)} = \frac{\lambda \zeta_M^b(N - 1)}{\zeta_M^b(N)} = \left(1 - \frac{|b_M'(N)|^2}{\zeta_M^b(N)} \right). \quad (46)$$

We can thus derive similar updates for the other reflection coefficients. Algorithm 3 is the resulting a-posteriori-based algorithm.

6. NORMALIZED EXTENDED RLS LATTICE ALGORITHM

A normalized lattice algorithm is an equivalent variant that replaces each pair of cross-reflection coefficient updates, that is, $\{\kappa_M^f(N), \kappa_M^b(N)\}$ and $\{\kappa_M^v(N), \kappa_M^b(N)\}$ by alternative updates based on single coefficients, which we denote by $\{\eta_M(N)\}$ and $\{\varphi_M(N)\}$. This is obtained by noting that these reflection coefficients are related to single parameters, that is, $\{\kappa_M^f(N), \kappa_M^b(N)\}$ related to $\delta_M(N)$ and $\{\kappa_M^v(N), \kappa_M^b(N)\}$ related to $\chi_M(N)$. The reflection coefficient $\kappa_M(N)$ is also replaced by $\omega_M(N)$.

6.1. Recursion for $\eta_M(N)$

We start by defining the coefficient

$$\eta_M(N) \triangleq \frac{\delta_M^*(N)}{\zeta_M^{b/2}(N) \zeta_M^{f/2}(N)} \quad (47)$$

along with the normalized prediction errors

$$\begin{aligned} b_M''(N) &\triangleq \frac{b_M(N)}{\gamma_M^{1/2}(N) \zeta_M^{b/2}(N)}, \\ f_M''(N) &\triangleq \frac{f_M(N)}{\bar{\gamma}_M^{1/2}(N) \zeta_M^{f/2}(N)}, \\ \bar{b}_M''(N) &\triangleq \frac{\bar{b}_M(N)}{\bar{\gamma}_M^{1/2}(N) \zeta_M^{b/2}(N)}, \\ v_M''(N) &\triangleq \frac{\nu_M(N)}{\gamma_M^{1/2}(N) \zeta_M^{v/2}(N)}. \end{aligned} \quad (48)$$

Now, referring to Algorithm 1, we substitute the updating equation for $\{\alpha_{M+1}(N)\}$ into the recursion for $\{\kappa_M^f(N)\}$. This yields

$$\begin{aligned} \kappa_M^f(N) &= \kappa_M^f(N - 1) \left(1 - |\bar{b}_M''(N)|^2 \right) \\ &\quad + \frac{f_M(N) \bar{b}_M^*(N)}{\zeta_M^b(N) \bar{\gamma}_M(N)}. \end{aligned} \quad (49)$$

<i>Initialization</i>
For $m = 0$ to M , set
μ is a small positive number
$\kappa_m(-1) = \kappa_m^b(-1) = \kappa_m^f(-1)$
$\gamma_m(-1) = \beta_m(-1) = 0$
$\zeta_m^f(-1) = \mu$
$\zeta_m^b(-1) = \pi_m - c_m^* \Pi_m c_m$
$\check{\zeta}_m^b(-1) = \check{\pi}_{m+1} - \check{c}_m^* \check{\Pi}_m \check{c}_m$
$\sigma_m = \lambda \zeta_m^b(-1) / \zeta_m^f(-1)$
$\chi_m(-1) = a_m \phi_m^* \Pi_m c_m + A_m$
$\kappa_m^b(-1) = \zeta_m^b(-1) \chi_m(-1)$
$\kappa_m^v(-1) = \chi_m^*(-1) / \zeta_m^b(-1)$
$\zeta_m^b(-1) = \zeta_{m+1}^b(-1)$
For $N \geq 0$, repeat
$\gamma_0(N) = \bar{\gamma}_0(N) = 1, \quad f_0(N) = b_0(N) = s(N)$
$v_0(N) = 0, \quad e_0(N) = d(N)$
For $m = 0$ to $M - 1$, repeat
$\check{\zeta}_m^b(N) = \sigma_m \bar{\gamma}_m(N) \zeta_m^f(N - 1)$
$\kappa_M^b(N) = \frac{\lambda \bar{\gamma}_M(N)}{\gamma_{M+1}(N-1)} \left(\kappa_M^b(N-1) + \frac{a_{M+1} \zeta_M^b(N-1) v_{M+1}^*(N-1) b_{M+1}(N-1)}{\gamma_{M+1}(N-1)} \right)$
$\bar{b}_m(N) = a_{m+1} b_{m+1}(N-1) + \kappa_m^b(N) v_{m+1}(N-1)$
$\zeta_m^f(N) = \lambda \zeta_m^f(N-1) + f_m(N) ^2 / \bar{\gamma}_m(N)$
$\zeta_m^b(N) = \lambda \zeta_m^b(N-1) + b_m(N) ^2 / \gamma_m(N)$
$\check{\zeta}_m^b(N) = \lambda \check{\zeta}_m^b(N-1) + \bar{b}_m(N) ^2 / \bar{\gamma}_m(N)$
$\gamma_{m+1}(N) = \gamma_m(N) - \frac{ b_m(N) ^2}{\zeta_m^b(N)}$
$\bar{\gamma}_{m+1}(N) = \bar{\gamma}_m(N) - \frac{ \bar{b}_m(N) ^2}{\check{\zeta}_m^b(N)}$
$\kappa_M^v(N) = \frac{\gamma_{M+1}(N)}{\lambda \gamma_M(N)} \left(\kappa_M^v(N-1) + \frac{a_M^* b_M^*(N) v_M(N)}{\gamma_M(N) \zeta_M^b(N-1)} \right)$
$\kappa_m^b(N) = \frac{\gamma_{m+1}(N)}{\gamma_m(N)} \left(\kappa_m(N-1) + \frac{f_m^*(N) \bar{b}_m(N)}{\bar{\gamma}_m(N) \zeta_m^f(N-1)} \right)$
$\kappa_m^f(N) = \frac{\bar{\gamma}_{m+1}(N)}{\bar{\gamma}_m(N)} \left(\kappa_m^f(N-1) + \frac{\bar{b}_m^*(N) f_m(N)}{\bar{\gamma}_m(N) \zeta_m^b(N-1)} \right)$
$\kappa_m(N) = \frac{\gamma_{m+1}(N)}{\gamma_m(N)} \left(\kappa_m(N-1) + \frac{b_m^*(N) e_m(N)}{\gamma_m(N) \zeta_m^b(N-1)} \right)$
$v_{m+1}(N) = -a_m^* v_m(N) + \kappa_m^v(N) b_m(N)$
$e_{m+1}(N) = e_m(N) - \kappa_m(N) b_m(N)$
$b_{m+1}(N) = \bar{b}_m(N) - \kappa_m^b(N) f_m(N)$
$f_{m+1}(N) = f_m(N) - \kappa_m^f(N) \bar{b}_m(N)$

ALGORITHM 3: The a-posteriori-based extended RLS lattice filter with direct reflection coefficients updates.

Multiplying both sides by the ratio $\zeta_M^{b/2}(N) / \zeta_M^{f/2}(N)$, we obtain

$$\eta_M(N) = \frac{\zeta_M^{b/2}(N)}{\zeta_M^{f/2}(N)} \kappa_M^f(N-1) \left(1 - |\bar{b}_M''(N)|^2 \right) + f_M''(N) \bar{b}_M''^*(N). \quad (50)$$

However, from the time-update recursion for $\zeta_M^b(N)$ and $\zeta_M^f(N)$, the following relations hold:

$$\zeta_M^{b/2}(N) = \frac{\lambda^{1/2} \zeta_M^{b/2}(N-1)}{\sqrt{1 - |\bar{b}_M''(N)|^2}}, \quad (51)$$

$$\zeta_M^{f/2}(N) = \frac{\lambda^{1/2} \zeta_M^{f/2}(N-1)}{\sqrt{1 - |f_M''(N)|^2}}.$$

Substituting these equations into (50), we obtain the desired time-update recursion for the first reflection coefficient:

$$\eta_M(N) = \eta_M(N-1) \sqrt{\left(1 - |\bar{b}_M''(N)|^2 \right) \left(1 - |f_M''(N)|^2 \right)} + f_M''(N) \bar{b}_M''^*(N). \quad (52)$$

This recursion is in terms of the errors $\{b_M''(N), f_M''(N)\}$. We thus need to determine order updates for these errors. Thus dividing the order-update equation for $b_{M+1}(N)$ by $\zeta_{M+1}^{b/2}(N) \gamma_{M+1}^{1/2}(N)$, we obtain

$$b_{M+1}''(N) = \frac{\bar{b}_M(N) - \kappa_M^b(N) f_M(N)}{\zeta_{M+1}^{b/2}(N) \gamma_{M+1}^{1/2}(N)}. \quad (53)$$

Using the order-update relation for $\zeta_M^b(N)$ we also have

$$\zeta_{M+1}^b(N) = \zeta_M^b(N) \left(1 - |\eta_M(N)|^2 \right). \quad (54)$$

In addition, the relation, for $\gamma_M(N)$,

$$\gamma_{M+1}(N) = \bar{\gamma}_M(N) - \frac{|f_M(N)|^2}{\zeta_M^f(N)} \quad (55)$$

can be written as

$$\gamma_{M+1}(N) = \bar{\gamma}_M(N) \left(1 - |f_M''(N)|^2 \right). \quad (56)$$

Therefore substituting (54) and (56) into (53), we obtain

$$b_{M+1}''(N) = \frac{\bar{b}_M''(N) - \eta_M^*(N) f_M''(N)}{\sqrt{\left(1 - |f_M''(N)|^2 \right) \left(1 - |\eta_M(N)|^2 \right)}}. \quad (57)$$

Similarly, using the order updates for $f_{M+1}(N)$, $\zeta_M^f(N)$, and $\bar{\gamma}_M(N)$ we obtain

$$f_{M+1}''(N) = \frac{f_M''(N) - \eta_M(N) \bar{b}_M''(N)}{\sqrt{\left(1 - |\bar{b}_M''(N)|^2 \right) \left(1 - |\eta_M(N)|^2 \right)}}. \quad (58)$$

6.2. Recursion for $\omega_M(N)$

In a similar vein, we introduce the normalized error

$$e''_M(N) \triangleq \frac{e_M(N)}{\gamma_M^{1/2}(N)\zeta_M^{1/2}(N)} \quad (59)$$

and the coefficient

$$\omega_M(N) \triangleq \frac{\rho_M(N)}{\zeta_M^{b/2}(N)\zeta_M^{1/2}(N)}. \quad (60)$$

Using the order update for $\zeta_M^{1/2}(N)$ and $\gamma_M(N)$, we can establish the following recursion:

$$e''_{M+1}(N) = \frac{e''_M(N) - \omega_M(N)b''_M(N)}{\sqrt{(1 - |b''_M(N)|^2)(1 - |\omega_M(N)|^2)}}. \quad (61)$$

To obtain a time update for $\omega_M(N)$, we first substitute the recursion for $e_{M+1}(N)$ into the time update for $\kappa_M(N)$. Then multiplying the resulting equation by the ratio $\zeta_M^{b/2}(N)/\zeta_M^{e/2}(N)$, and using the time updates for $\zeta_M^b(N)$ and $\zeta_M(N)$, we obtain

$$\begin{aligned} \omega_M(N) &= \sqrt{(1 - |b''_M(N)|^2)(1 - |e''_M(N)|^2)}\omega_M(N - 1) \\ &\quad + b''_{M*}(N)e''_M(N). \end{aligned} \quad (62)$$

Note that when $\bar{b}''_M(N) = b''_M(N - 1)$, the recursions derived so far collapse to the well-known FIR normalized RLS lattice algorithm. For general structures, however, we need to derive a recursion for the normalized variable $\bar{b}''_M(N)$ as well. This can be achieved by normalizing the order update for $\bar{b}_M(N)$:

$$\bar{b}_M(N) = \frac{a_{M+1}b_{M+1}(N - 1) + \kappa_M^b(N)v_{M+1}(N - 1)}{\zeta_M^{b/2}(N)\bar{\gamma}_M^{1/2}(N)}. \quad (63)$$

In order to simplify this equation, we need to relate $\zeta_M^b(N)$ to $\zeta_{M+1}^b(N - 1)$ and $\bar{\gamma}_M(N)$ to $\gamma_{M+1}(N - 1)$. Recalling the alternative update for $\zeta_M^b(N)$:

$$\begin{aligned} \zeta_M^b(N) &= |a_{M+1}|^2\zeta_{M+1}^b(N - 1) \\ &\quad + \frac{|\chi_{M+1}(N - 1)|^2}{\zeta_{M+1}^v(N - 1)}, \end{aligned} \quad (64)$$

we get

$$\zeta_M^b(N) = \zeta_{M+1}^b(N - 1)(|a_{M+1}|^2 + |\varphi_{M+1}(N - 1)|^2), \quad (65)$$

where we have defined the reflection coefficient

$$\varphi_M(N) \triangleq \frac{\chi_M(N)}{\zeta_M^{b/2}(N)\zeta_M^{v/2}(N)}. \quad (66)$$

In order to relate $\{\bar{\gamma}_M(N), \gamma_{M+1}(N - 1)\}$, we resort to the alternative relation of Algorithm 1:

$$\bar{\gamma}_M(N) = \gamma_{M+1}(N - 1) + \zeta_M^b(N)|v_{M+1}(N - 1)|^2 \quad (67)$$

which can be written as

$$\bar{\gamma}_M(N) = \gamma_{M+1}(N - 1)(1 + |v''_{M+1}(N - 1)|^2). \quad (68)$$

Substituting (65) and (68) into (63), we obtain

$$\begin{aligned} \bar{b}''_M(N) &= \frac{a_{M+1}b''_{M+1}(N - 1) + \varphi_{M+1}(N - 1)v''_{M+1}(N - 1)}{\sqrt{(1 + |v''_{M+1}(N - 1)|^2)(|a_{M+1}|^2 + |\varphi_{M+1}(N - 1)|^2)}}. \end{aligned} \quad (69)$$

This equation requires an order update for the normalized quantity $v''_M(N)$. From the order update for $v_M(N)$, we can write

$$v''_{M+1}(N) = \frac{-a_M^*v_M(N) + \kappa_M^v(N)b_M(N)}{\zeta_{M+1}^{v/2}(N)\gamma_{M+1}^{1/2}(N)}. \quad (70)$$

Similarly to (63), we need to relate $\{\zeta_{M+1}^v(N), \gamma_{M+1}(N)\}$ with $\{\zeta_M^v(N), \gamma_M(N)\}$. Thus recall that these quantities satisfy the following order updates:

$$\begin{aligned} \zeta_{M+1}^v(N) &= |a_M|^2\zeta_M^v(N) + \frac{|\chi_M(N)|^2}{\zeta_M^b(N)}, \\ \gamma_{M+1}(N) &= \gamma_M(N) - \frac{|b_M(N)|^2}{\zeta_M^b(N)}, \end{aligned} \quad (71)$$

which lead to the following relations:

$$\begin{aligned} \zeta_{M+1}^v(N) &= \zeta_M^v(N)(|a_M|^2 + |\varphi_M(N)|^2), \\ \gamma_{M+1}(N) &= \gamma_M(N)(1 - |b''_M(N)|^2). \end{aligned} \quad (72)$$

Taking the square root on both sides of (72) and substituting into (70), we get

$$v''_{M+1}(N) = \frac{-a_M^*v''_M(N) + \varphi_M^*(N)b''_M(N)}{\sqrt{(1 - |b''_M(N)|^2)(|a_M|^2 + |\varphi_M(N)|^2)}}. \quad (73)$$

6.3. Recursion for $\varphi_M(N)$

This is the only remaining recursion, which is defined via (66). To derive an update for it, we proceed similarly to the recursions for $\{\eta_M(N), \omega_M(N)\}$. First we substitute the update for $v_{M+1}(N)$ into the update for $\kappa_M^v(N)$ in Algorithm 1. This gives

$$k_M^v(N) = \lambda^{-1} \left(1 - |b_M''(N)|^2 \right) k_M^v(N-1) + a_M^* \frac{b_M^{*'}(N) v_M^v(N)}{\zeta_M^b(N)}. \quad (74)$$

Then, multiplying the above equation by $\zeta_M^{b/2}(N)/\zeta_M^{v/2}(N)$ and using the fact that

$$\zeta_M^{b/2}(N) = \frac{\lambda^{1/2} \zeta_M^{b/2}(N-1)}{\sqrt{1 - |b_M''(N)|^2}}, \quad (75)$$

$$\zeta_M^{v/2}(N) = \frac{\lambda^{-1/2} \zeta_M^{v/2}(N-1)}{\sqrt{1 + |v_M^v(N)|^2}}$$

(see the equalities in (43) and (46)), we get

$$\varphi_M(N) = \sqrt{\left(1 + |v_M^v(N)|^2 \right) \left(1 - |b_M''(N)|^2 \right)} \varphi_M(N-1) + a_M^* b_M^{*'}(N) v_M^v(N). \quad (76)$$

Algorithm 4 is the resulting normalized extended RLS lattice algorithm. For compactness of notation and in order to save in computations, we introduced the variables

$$\left\{ r_M^b(N), r_M^f(N), r_M^e(N), r_M^v(N), r_M^\varphi(N), r_M^{\bar{b}}(N), r_M^\eta(N), r_M^\omega(N) \right\}. \quad (77)$$

Note that the normalized algorithm returns the normalized least-squares residual $e_{M+1}''(N)$. The original error $e_{M+1}(N)$ can be easily recovered, since the normalization factor can be computed recursively by

$$\zeta_{M+1}^{1/2}(N) \gamma_{M+1}^{1/2}(N) = r_M^b(N) r_M^\omega(N) \zeta_M^{1/2}(N) \gamma_M^{1/2}(N). \quad (78)$$

7. ARRAY-BASED LATTICE ALGORITHM

We now derive another equivalent lattice form, albeit one that is described in terms of compact arrays.

To arrive at the array form, we first define the following quantities:

$$q_M^b(N) \triangleq \frac{\delta_M(N)}{\zeta_M^{b/2}(N)}, \quad q_M^f(N) \triangleq \frac{\delta_M^*(N)}{\zeta_M^{f/2}(N)}, \quad (79)$$

$$q_M^{\bar{b}}(N) \triangleq \frac{\chi_M(N)}{\zeta_M^{v/2}(N)}, \quad q_M^v(N) \triangleq \frac{\chi_M^*(N)}{\zeta_M^{b/2}(N)}.$$

Initialization
For $m = 0$ to M , set
μ is a small positive number
$\eta_m(-1) = \omega_m(-1) = b_m''(-1) = v_m^v(-1) = 0$
$\zeta_m^b(-1) = \pi_m - c_m^* \Pi_m c_m$
$\varphi_{m+1}(-1) = \sqrt{\frac{\bar{\pi}_{m+1} - \bar{c}_m^* \Pi_m \bar{c}_{m+1}}{\zeta_{m+1}^b(-1)}} (a_m \varphi_m^* \Pi_m c_m + A_m)$
For $N \geq 0$, repeat
$\zeta_0^b(N) = \lambda \zeta_0^b(N-1) + u(N) ^2$
$\zeta_0(N) = \lambda \zeta_0(N-1) + d(N) ^2$
$b_0^v(N) = f_0''(N) = u(N)/\zeta_0^{b/2}(N)$
$e_0''(N) = d(N)/\zeta_0^{1/2}(N)$
$v_0^v(N) = 0$
For $m = 0$ to $M-1$, repeat
$r_m^b(N) = \sqrt{1 - b_m''(N) ^2}$, $r_m^f(N) = \sqrt{1 - f_m''(N) ^2}$
$r_m^e(N) = \sqrt{1 - e_m''(N) ^2}$, $r_m^v(N) = \sqrt{1 + v_m^v(N) ^2}$
$\varphi_m(N) = r_m^v(N) r_m^b(N) \varphi_m(N-1) + a_m^* b_m^{*'}(N) v_m^v(N)$
$r_m^\varphi(N) = \sqrt{ a_m ^2 + \varphi_m(N) ^2}$
$\bar{b}_m^v(N) = \frac{a_{M+1} b_{m+1}''(N-1) + \varphi_{m+1}(N-1) v_{m+1}^v(N-1)}{r_{m+1}^v(N-1) r_{m+1}^\varphi(N-1)}$
$r_m^{\bar{b}}(N) = \sqrt{1 - \bar{b}_m^v(N) ^2}$
$\eta_m(N) = r_m^{\bar{b}}(N) r_m^f(N) \eta_m(N-1) + f_m''(N) \bar{b}_m^v(N)$
$r_m^\eta(N) = \sqrt{1 - \eta_m(N) ^2}$
$\omega_m(N) = r_m^b(N) r_m^e(N) \omega_m(N-1) + b_m^{*'}(N) e_m''(N)$
$r_m^\omega(N) = \sqrt{1 - \omega_m(N) ^2}$
$v_{m+1}^v(N) = \frac{1}{r_m^b(N) r_m^\varphi(N)} [-a_m^* v_m^v(N) + \varphi_m^*(N) b_m''(N)]$
$e_{m+1}''(N) = \frac{1}{r_m^b(N) r_m^\eta(N)} [e_m''(N) - \omega_m(N) b_m''(N)]$
$b_{m+1}^v(N) = \frac{1}{r_m^f(N) r_m^\eta(N)} [\bar{b}_m^v(N) - \eta_m^*(N) f_m''(N)]$
$f_{m+1}''(N) = \frac{1}{r_m^b(N) r_m^\eta(N)} [f_m''(N) - \eta_m(N) \bar{b}_m^v(N)]$

ALGORITHM 4: Normalized extended RLS lattice filter.

The second step is to rewrite all the recursions in Algorithm 1 in terms of these quantities, and in terms of the angle normalized prediction errors $\{b_M^v(N), e_M^v(N), v_M^v(N), \bar{b}_M^v(N)\}$ defined before, for example,

$$\chi_M(N) = \chi_M(N-1) + a_M v_M^{*'}(N) b_M(N),$$

$$\zeta_M^v(N) = \lambda^{-1} \zeta_M^v(N-1) - |v_M^v(N)|^2, \quad (80)$$

$$\zeta_M^b(N) = \lambda \zeta_M^b(N-1) + |b_M^v(N)|^2.$$

The third step is to implement a unitary (Givens) transformation Θ_M that lower triangularizes the following pre-array of numbers:

$$\underbrace{\begin{bmatrix} \lambda^{1/2} \zeta_M^{b/2}(N-1) & b_M^{*'}(N) \\ \lambda^{-1/2} q_M^{v*}(N-1) & a_M v_M^{*'}(N) \end{bmatrix}}_A \Theta_M = \underbrace{\begin{bmatrix} m & 0 \\ n & p \end{bmatrix}}_B \quad (81)$$

for some $\{m, n, p\}$. The values of the resulting $\{m, n, p\}$ can be determined from the equality $A\Theta_M\Theta_M^*A^* = AA^* = BB^*$, which gives

$$m = \zeta_M^{b/2}(N), \quad n = q_M^{v*}(N), \quad p = v_{M+1}^{*}(N). \quad (82)$$

Similarly, we can implement a J -unitary (hyperbolic) transformation $\Theta_M^{\bar{b}}$ that lower triangularizes the following prearray of numbers:

$$\underbrace{\begin{bmatrix} \lambda^{-1/2}\zeta_M^{v/2}(N-2) & v_{M+1}^{*}(N-1) \\ \lambda^{1/2}q_M^{b*}(N-1) & -a_{M+1}b_{M+1}^{*}(N-1) \end{bmatrix}}_C \Theta_M^{\bar{b}} = \underbrace{\begin{bmatrix} m' & 0 \\ n' & p' \end{bmatrix}}_D \quad (83)$$

for some $\{m', n', p'\}$, which can be determined from the equality $C\Theta_M^{\bar{b}}\Theta_M^{\bar{b}*}C^* = CJC^* = DD^*$. This gives

$$m' = \zeta_M^{v/2}(N-1), \quad n' = q_M^{\bar{b}*}(N), \quad p' = \bar{b}_M^{*}(N). \quad (84)$$

Proceeding similarly we can derive two additional array transformations, all of which are listed in Algorithm 5. The matrices $\{\Theta_M^f(N), \Theta_M^b(N)\}$ are 2×2 unitary (Givens) transformations that introduce the zero entries in the postarrays at the desired locations. We illustrate the m th array lattice section that corresponds to this algorithm in Figure 3.

8. SIMULATION RESULTS

We have performed several simulations in Matlab in order to verify the behavior of the proposed lattice variants under finite-precision arithmetic. Under infinite precision, all lattice filters are equivalent. However, unlike what is normally expected from the corresponding standard FIR lattice filter variants, which tend to be more reliable in finite precision, we observed that the new algorithms exhibit some unstable patterns when compared with the standard lattice recursions of Algorithm 1, in the sense that at some point the algorithms diverge or possibly achieve a much higher MSE.

In the sequel, we will present some simulation results for the algorithms obtained in this paper. We have tested the algorithms over 500 runs, for an exact system identification of a 5-tap orthonormal basis. The noise floor was fixed at -50 dB.

We have observed different behaviors throughout the several scenarios set for testing. In order to characterize their performance up to a certain extent, we have selected some of these settings, which we believe to be the most relevant ones for this purpose.

Experiment 1 (comparison among all algorithms under Matlab precision). Here we have set the forgetting factor $\lambda = 0.99$, a typical value, for all the recursions. As a result, we have observed (Figure 4) the best performance for the *a posteriori error-feedback* version, followed by the *a priori error-feedback* recursion exhibiting a slightly higher MSE.

Initialization

For $m = 0$ to $M - 1$, set

μ is a small positive number

$\zeta_m^f(-1) = \mu$

$\zeta_m^b(-1) = \pi_m - c_m^* \Pi_m c_m$

$\check{\zeta}_m^b(-1) = \check{\pi}_{m+1} - \check{c}_m^* \check{\Pi}_m \check{c}_m$

$\zeta_m^{\bar{b}}(-1) = \zeta_{m+1}^b(-1) \chi_m(-1) = a_m \phi_m^* \Pi_m c_m + A_m$

$q_m(-1) = q_m^b(-1) = q_m^f(-1) = v_m'(-1) = b_m'(-1) = 0$

$q_m^{\bar{b}}(-1) = \chi_m(-1) / \zeta_m^{v/2}(-1)$

$q_m^v(-1) = \chi_m^*(-1) / \zeta_m^{b/2}(-1)$

For $N \geq 0$, repeat

$\gamma_0^{1/2}(N) = 1$

$e_0'(N) = d(N)$

$f_0'(N) = b_0'(N) = u(N)$

$v_0'(N) = 0$

For $m = 0$ to $M - 1$, repeat

$$\begin{bmatrix} \lambda^{-1/2}\zeta_m^{v/2}(N-2) & v_{m+1}^{*}(N-1) \\ \lambda^{1/2}q_m^{b*}(N-1) & -a_{m+1}b_{m+1}^{*}(N-1) \end{bmatrix} \Theta_m^{\bar{b}}(N)$$

$$= \begin{bmatrix} \zeta_m^{v/2}(N-1) & 0 \\ q_m^{\bar{b}*}(N) & \bar{b}_m^{*}(N) \end{bmatrix}$$

$$\begin{bmatrix} \lambda^{1/2}\zeta_m^{b/2}(N-1) & b_m^{*}(N) \\ \lambda^{-1/2}q_m^{v*}(N-1) & a_m v_m^{*}(N) \\ \lambda^{1/2}q_m^{d*}(N-1) & e_m^{*}(N) \\ 0 & \gamma_m^{1/2}(N) \end{bmatrix} \Theta_m(N)$$

$$= \begin{bmatrix} \zeta_m^{b/2}(N) & 0 \\ q_m^{v*}(N) & v_{m+1}^{*}(N) \\ q_m^{d*}(N) & e_{m+1}^{*}(N) \\ \times & \gamma_{m+1}^{1/2}(N) \end{bmatrix}$$

$$\begin{bmatrix} \lambda^{1/2}\zeta_m^{f/2}(N-1) & f_m^{*}(N) \\ \lambda^{1/2}q_m^{f*}(N-1) & \bar{b}_m^{*}(N) \end{bmatrix} \Theta_m^f(N)$$

$$= \begin{bmatrix} \zeta_m^{f/2}(N) & 0 \\ q_m^{f*}(N) & b_{m+1}^{*}(N) \end{bmatrix}$$

$$\begin{bmatrix} \lambda^{1/2}\zeta_m^{\bar{b}/2}(N-1) & \bar{b}_m^{*}(N) \\ \lambda^{1/2}q_m^{b*}(N-1) & f_m^{*}(N) \end{bmatrix} \Theta_m^b(N)$$

$$= \begin{bmatrix} \zeta_m^{\bar{b}/2}(N) & 0 \\ q_m^{b*}(N) & f_{m+1}^{*}(N) \end{bmatrix}$$

ALGORITHM 5: The array-based extended RLS lattice filter.

Although the normalized recursion appears to have an even higher MSE and the QR lattice algorithm does not converge, we observed that their behavior changes when $\lambda = 1$. In order to observe their behavior more closely, we tested each algorithm separately in fixed-point arithmetic, as shown next.

Experiment 2 (a priori error-feedback algorithm for different values of λ). This is shown in Figure 5. In these simulations, we have arbitrarily limited the number of fixed-point quantization steps to 16 bits. Unlike what is observed in experiment 1, this algorithm diverges at some point depending on the value of λ used.

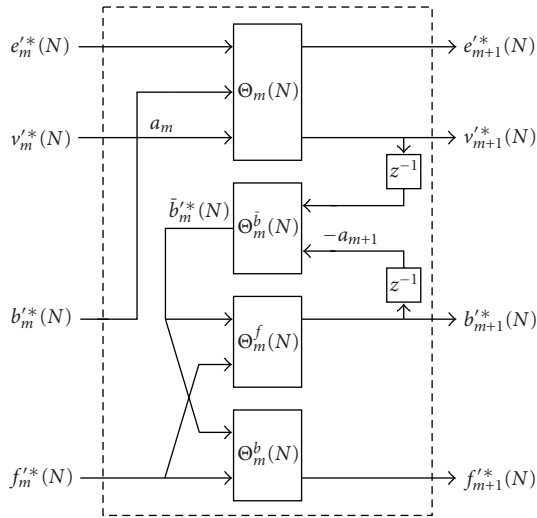


FIGURE 3: A lattice section in array form.

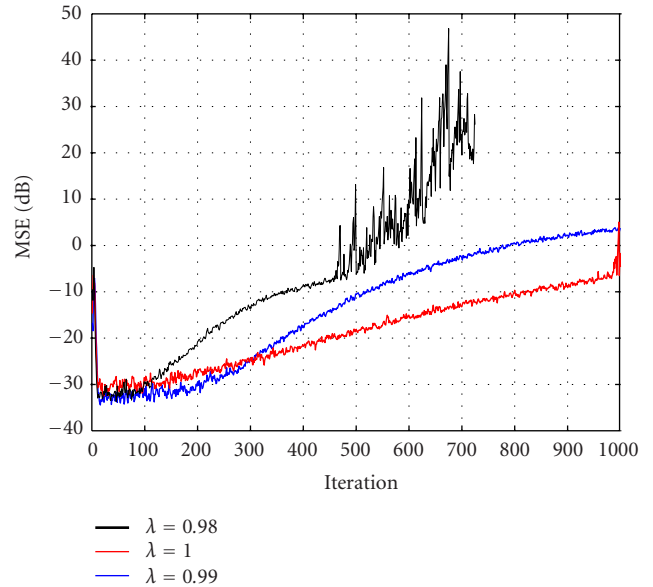


FIGURE 5: A priori error-feedback algorithm for different values of $\lambda = 1, 0.99,$ and 0.98 .

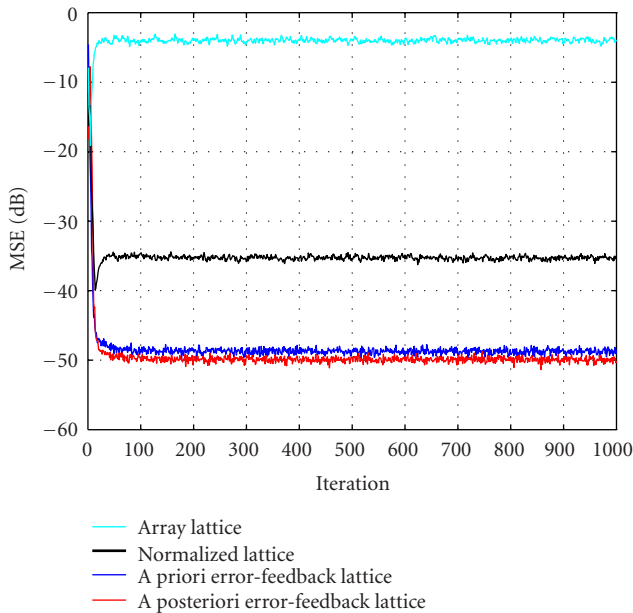


FIGURE 4: Comparison among all algorithms under Matlab precision.

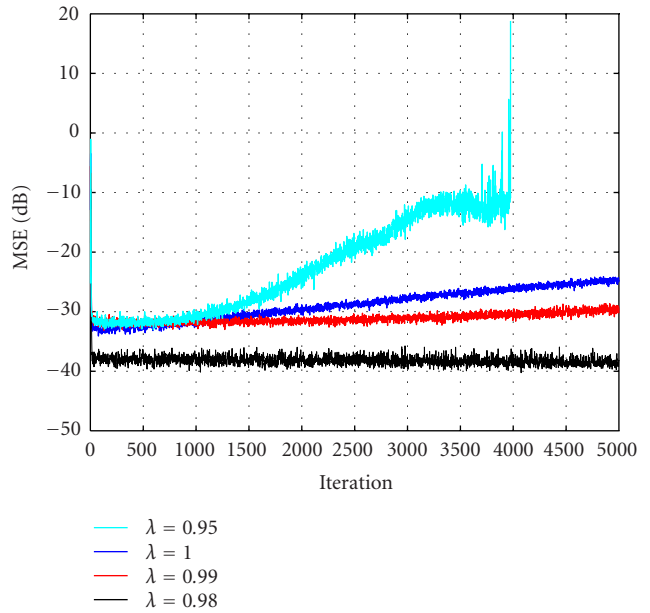


FIGURE 6: A posteriori error-feedback algorithm for $\lambda = 1, 0.99,$ and 0.95 .

Experiment 3 (a posteriori error-feedback algorithm for different values of λ). This is shown in Figure 6. Similarly to experiment 2, we have kept the quantization steps to 16 bits. Although the scenario of $\lambda = 0.98$ seems to be more steady, we still observed divergence when running the experiment over 10 000 data samples.

Experiment 4 (performance for 24-bit quantization and for $\lambda = 1$). In experiment 1, these algorithms appeared to have a higher MSE compared to the error-feedback versions, and

does not seem to converge to the noise floor. However, we have noticed that in the case where $\lambda = 1$, they present some convergence up to a certain instant (depending on how big the number of quantization steps is), and then diverge in two different ways. This is illustrated in Figure 7, for a 24-bit fixed-point quantization. For a less number of bits, the performance becomes worse.

In summary, all lattice variants become unstable, except for the standard equations of Algorithm 1, of which the

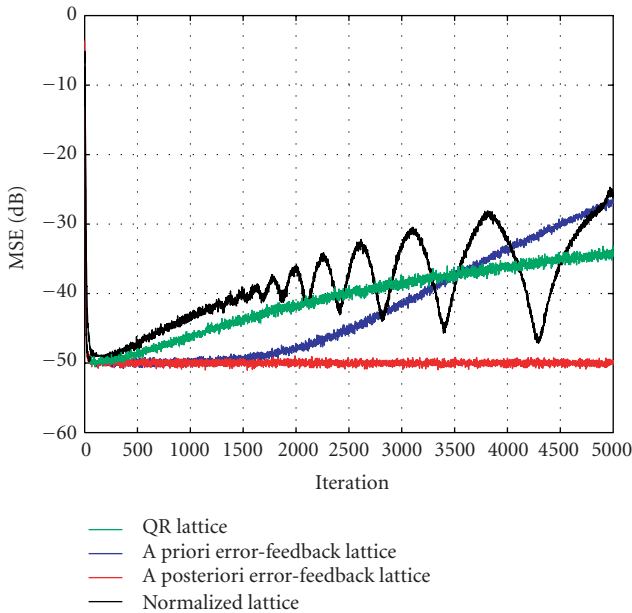


FIGURE 7: Performance of the algorithms for $\lambda = 1$, and 24-bit quantization.

corresponding MSE curve presents unstable behavior only for $\lambda = 1$. Figure 8 illustrates this fact for $\lambda = 1, 0.9$, and 0.85 , considering a 5-tap orthonormal model.

9. BACKWARD CONSISTENCY AND MINIMALITY ISSUES

The key structural problem for the unstable behavior observed in all the above algorithms relies on the nonminimality of the state vector in the backward prediction portion of each algorithm. In order to elaborate on this point with some depth, we will briefly review the concepts of *minimality* and *backward consistency* for FIR structures ([15, 16, 17, 18]) and extend these arguments to the algorithms of this paper.

Error analysis in fast RLS algorithms is performed in the prediction portion of the recursions, since the flow of the information required for the optimal least-squares solution is one way to the joint process estimation section. The prediction part of the algorithm is a nonlinear mapping of the form

$$s_i = \mathcal{T}[s_{i-1}, u(i)], \tag{85}$$

where s_i denotes the state vector that contains the variables propagated by the underlying algorithm. In finite-precision implementations, however, the actual mapping propagates the perturbed state \hat{s}_i , that is,

$$\hat{s}_i = \mathcal{T}[s_{i-1}, u(i)] + \delta_i, \tag{86}$$

where δ_i is result of quantization. In this case, one's primary goal is to show exponential stability of this system, that is, to

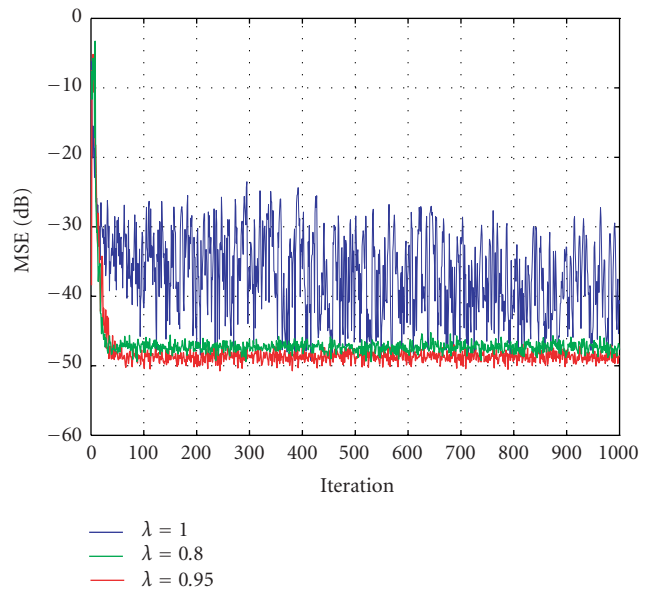


FIGURE 8: Learning curve for the standard extended lattice recursions quantized in 16 bits for different values of λ and for an order-5 filter.

show (or not) that the influence of such perturbation should decay sufficiently fast to zero as $i \rightarrow \infty$. Thus let \mathcal{S}_i denote the set of all state values $\{s_i\}$ for which the mapping (86) is exponentially stable. This set includes all state values that can be reached in exact arithmetic, as the input $\{u(i), u(i-1), \dots\}$ varies over all realizations that are persistently exciting (we will return to the persistency of excitation issue shortly, considering the general orthonormal basis studied here). Clearly, the state error $\bar{s}_i = s_i - \hat{s}_i$ will remain bounded provided that the system (86) is exponentially stable for all states s_i and the perturbation δ_i does not push \hat{s}_i outside \mathcal{S}_i . Now, in order to fully understand round-off error effect in a given algorithm, one must consider three aspects in its analysis:

- (1) *error generation*, that is, the properties of the round-off error δ_i ;
- (2) *error accumulation*, that is, how the overall state error \bar{s}_i is affected by the intermediate errors generated at different time instants;
- (3) *error propagation*, in the sense that it is assumed that from a certain time instant, no more round-off errors are made, and the propagation of the accumulated errors from that point onward is observed.

Since it is not our intention to embark on a detailed error analysis of these algorithms, we will consider only error propagation stability, which is equivalent to exponential stability of the time recursions. (A conventional stability analysis of an algorithm is usually difficult to pursue, due to the nonlinear nature of system \mathcal{T} . This can be accomplished, however, via local linearization and Lyapunov methods, although this requires considerable effort).

A richer approach to the stability problem relies on checking the so-called *backward consistency* property, that

is, if a computed solution (with round-off errors) is the exact solution to a perturbed problem in exact arithmetic. In other words, denote by $\hat{\mathcal{S}}_i$ the set of all state vectors that are reachable in finite-precision arithmetic, which vary according to the implementations of the algorithm recursions (that is, the effect of word length plus rounding/truncation). Then, if $\hat{\mathcal{S}}_i \subset \mathcal{S}_i$, the algorithm is said to be *backward consistent* in this context.

Now, an algorithm is said to be *nonminimal*, if the elements of the state vector s_i can be expressed in a reduced dimension with no loss of information, that is, loosely speaking, when the recursions that constitute the algorithm propagate redundancy. In this case, these redundant components can be expressed in terms of constraints of the form

$$f(s_i) = 0, \quad \forall i \text{ and } \{u(i)\} \quad (87)$$

which defines a manifold. In this case, there always exist local perturbations that push s_i outside this manifold and therefore out of the stability region \mathcal{S}_i .

It is proved in [15, 16, 17, 18], for fast FIR RLS recursions, that the minimal dimension of s_i is $2M + 1$, for all $i \geq 2M$. In this case, none of the FIR counterparts of the algorithms presented in this paper is minimal, which propagate around $5M$ variables (this does not mean that stable propagation of the variables cannot be proved without resorting to backward consistency issues, as for example in the case of the a priori error-feedback FIR lattice filter [19]).

Now, these minimal components for the FIR case can be established, once the connection with the minimal components of fast transversal filters of all least-squares orders is recognized, thus resulting in a $2M + 1$ minimal dimension (see [17]). We have shown in [3], for the general orthonormal basis of this paper, that the same defining components of the minimum state vector in a fast transversal FIR algorithm also define the minimum components of an orthonormality-based FTF,³ therefore, using the arguments of [17], one can conclude that this holds similarly for the order-recursive algorithms of this paper, resulting in $2M + 1$ minimal parameters. The nonminimal character of the above recursions can be intuitively seen, by following their derivation. It is like solving fast transversal least-squares of all orders, except that the necessity of calculating the augmented Kalman gain vector $\check{g}_{M+1,N}$ (in order to propagate $g_{M,N}$) on which the FTF is based, is replaced by the use of the augmented scalar $\bar{b}_{M+1,N}$ (in order to propagate $b_M(N)$), on which the lattice recursions are based.

Clearly, the extended lattice algorithms of this paper are nonminimal; they propagate an additional redundancy that eventually leads to divergence. Consider for instance, the array-based lattice filter obtained. Besides the $5M$ variables propagated in the forward and backward prediction section,

namely, $\{\xi_m^f(N), \xi_m^b(N), q_m^f(N), q_m^b(N), \bar{b}'_m(N)\}$, it also needs $\{\xi_m^v(N), q_m^b(N), v'_{M+1}\}$, which are updated by Θ^b . Now, note that in the FIR case, even though minimality of these recursions is violated, error propagation in all variables is exponentially stable, since the prearray is scaled by $\sqrt{\lambda}$. The analysis in this case is sufficient for an individual section, since any given lattice section is constituted only by lower-order ones. For the extended array lattice of this paper, however, two facts contribute for divergence. First, variables are further propagated via an unstable mode, that is, $\lambda^{-1/2}$. Second, it makes use of hyperbolic rotations, which are well known to be naturally unstable (unless some care is taken). We recall that in the case of FIR models, only circular, and therefore stable rotations are needed.

In the case of error-feedback algorithms, besides non-minimality, the presence of λ^{-1} in the recursion for the reflection coefficient $\kappa_M^v(N)$ contributes similarly to divergence. This behavior can also be observed in the standard lattice algorithm if one attempts to propagate the minimum cost $\zeta_M^b(N)$ via its inverse $\zeta_M^v(N)$, which also depends on λ^{-1} . The existence of such recursion is also a source of instability in fast fixed-order RLS counterparts. For the normalized lattice algorithm, because all prediction errors are normalized, these recursions become independent of the forgetting factor (except for $\{\zeta_0(N), \zeta_0^b(N)\}$ in the initial step). This, however, eliminates the need for recursion $\zeta_m^b(N) = \sigma_m \bar{y}_m(N) \zeta_m^f(N - 1)$, an enforcing relation that represents a source of good numerical conditioning for the algorithm. This relation helps in reducing the redundant variables in fast RLS recursions and is one of the relations forming the manifold of \mathcal{S}_i in such recursions (as observed for the FTF algorithm in [15, 16] in the case of FIR models).⁴ This relation has been further extended to the general orthonormal model of [3] and has been used in the standard and it turns out that this is also the case for the lattice recursions, since we have shown that this relation holds for every order- M LS problem.

It is important to add that the fast array algorithm considered in this paper (also called a *rotation-based* algorithm) is one among a few other fast QR-based recursions. It computes the forward and backward prediction errors in ascending order, leading to the conventional lattice networks studied in this paper. Still, other QR variants are also possible, one of them in fact resulting in a minimal realization for FIR structures [17]; it computes the forward prediction errors in ascending order, but the backward prediction errors in descending order. This is an important case of study, whose extension to the orthonormal basis case will be pursued elsewhere.

9.1. Persistency of excitation

One must distinguish between stability due to the algorithm structure and ill-conditioning of the underlying input data

³In the FTF-FIR case, the set \mathcal{S}_i is represented by the variables that satisfy a certain spectral factorization with respect to the FIR basis functions. This also holds true for the extended basis of this paper, except that spectral factorization is performed with respect to the orthonormal basis.

⁴Note that even though we are reducing the number of reflection coefficients in half, we end up not using this recursion, since variables are normalized by their energies.

matrix. Ill-conditioned data may push the states x_i outside the interior of \mathcal{S}_i , undermining the exponential stability of the recursions. The question then is whether the change of basis would affect the numerical error properties in the above lattice extensions.

Now, one of the main purposes of using orthonormal bases lies in the well-studied good numerical conditioning offered by such structures (where input conditioning remains unaltered). Note that this is not the case for an arbitrary model realization as a fixed denominator, of partial-fraction representation as pointed out in [2]. Consider the vector of basis functions associated with the orthonormal model:

$$\mathcal{B}(z) = [\mathcal{B}_0(z) \ \mathcal{B}_1(z) \ \cdots \ \mathcal{B}_{M-1}(z)]^T. \quad (88)$$

Under the assumption of stationarity of the input sequence $u(n)$, the input correlation matrix associated with these basis functions is given by

$$R_{\mathcal{B}} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{B}(e^{j\omega}) \mathcal{B}^*(e^{j\omega}) S_u(e^{j\omega}) d\omega, \quad (89)$$

where $S_u(e^{j\omega})$ is the power spectral density of $u(n)$. Now the conditioning of the (linear) estimation problem associated with the orthonormal model is related to the condition number of $R_{\mathcal{B}}$, namely, $k(R_{\mathcal{B}})$, which shows that orthonormality of the basis functions basically leaves the condition number unaltered when passing through such model. Therefore we can say that for the same condition number, the extended lattice recursions behave worse than their tapped-delay-line lattice counterparts.

10. CONCLUSION

In this work, we developed several lattice forms for RLS adaptive filters based on general orthonormal realizations. One technique is based on propagating the reflection coefficients in time. A second form is based on propagating a fewer number of normalized reflection coefficients. A third form is based on propagating angle-normalized quantities via unitary and J -unitary rotations. Even though the algorithms are all theoretically equivalent, they differ in computational cost and in robustness to finite-precision effects. The new algorithms, besides nonminimality, present unstable modes as well as hyperbolic rotations, so that the well-known good numerical properties observed in the class of FIR models no longer exist for the extended fast recursions derived. In this context, the standard lattice recursions of Algorithm 1 represent up to now the most numerically reliable algorithm for this class of input data structures.

We remark that the development of the above recursions represent an initial step towards further refinement of these algorithms and it is not our purpose to provide all answers on these extended lattice filter variants in this presentation. Although our presentation lacks a more precise analysis of the numerical behavior of these algorithms, we believe that the

arguments in Section 9 suffice as a preliminary explanation for the unstable behavior observed in all lattice variants. As future works, we will look into the numerical issues of these algorithms in detail as well as pursue a minimal fast QR realization similarly to [17] for FIR models.

ACKNOWLEDGMENT

This research is partially funded by CNPq, FAPERJ, and the José Bonifácio Foundation, Brazil.

REFERENCES

- [1] R. Merched, "Extended RLS lattice adaptive filters," *IEEE Trans. Signal Processing*, vol. 51, no. 9, pp. 2294–2309, 2003.
- [2] P. Heuberger, B. Ninness, T. Oliveira e Silva, P. Van den Hof, and B. Wahlberg, "Modeling and identification with orthogonal basis functions," in *Proc. 36th IEEE CDC Pre-Conference Workshop, number 7*, San Diego, Calif, USA, December 1997.
- [3] R. Merched and A. H. Sayed, "Extended fast fixed-order RLS adaptive filters," *IEEE Trans. Signal Processing*, vol. 49, no. 12, pp. 3015–3031, 2001.
- [4] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, "Computationally efficient QR decomposition approach to least squares adaptive filtering," *IEE Proceedings*, vol. 138, no. 4, pp. 341–353, 1991.
- [5] P. A. Regalia and M. G. Bellanger, "On the duality between fast QR methods and lattice methods in least squares adaptive filtering," *IEEE Trans. Signal Processing*, vol. 39, no. 4, pp. 879–891, 1991.
- [6] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias, *Advanced Digital Signal Processing*, MacMillan Publishing, New York, NY, USA, 1992.
- [7] P. Strobach, *Linear Prediction Theory*, Springer-Verlag, Berlin, Germany, 1990.
- [8] B. Yang and J. F. Bohme, "Rotation-based RLS algorithms: unified derivations, numerical properties, and parallel implementations," *IEEE Trans. Signal Processing*, vol. 40, no. 5, pp. 1151–1167, 1992.
- [9] R. Merched and A. H. Sayed, "RLS-Laguerre lattice adaptive filtering: error-feedback, normalized, and array-based algorithms," *IEEE Trans. Signal Processing*, vol. 49, no. 11, pp. 2565–2576, 2001.
- [10] J. W. Davidson and D. D. Falconer, "Reduced complexity echo cancellation using orthonormal functions," *IEEE Trans. Circuits and Systems*, vol. 38, no. 1, pp. 20–28, 1991.
- [11] L. Salama and J. E. Cousseau, "Efficient echo cancellation based on an orthogonal adaptive IIR realization," in *Proc. SBT/IEEE International Telecommunications Symposium (ITS '98)*, vol. 2, pp. 434–437, São Paulo, Brazil, August 1998.
- [12] J. J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Magazine*, vol. 6, no. 2, pp. 4–21, 1989.
- [13] P. A. Regalia, *Adaptive IIR Filtering in Signal Processing and Control*, Marcel Dekker, New York, NY, USA, 1995.
- [14] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Mag.*, vol. 11, no. 3, pp. 18–60, 1994.
- [15] D. T. M. Slock, "backward consistency concept and round-off error propagation dynamics in recursive least-squares algorithms," *Optical Engineering*, vol. 31, no. 06, pp. 1153–1169, 1992.
- [16] P. A. Regalia, "Numerical stability issues in fast least-squares adaptation algorithms," *Optical Engineering*, vol. 31, no. 06, pp. 1144–1152, 1992.

- [17] P. A. Regalia, "Numerical stability properties of a QR-based fast least squares algorithm," *IEEE Trans. Signal Processing*, vol. 41, no. 6, pp. 2096–2109, 1993.
- [18] P. A. Regalia, "Past input reconstruction in fast least-squares algorithms," *IEEE Trans. Signal Processing*, vol. 45, no. 9, pp. 2231–2240, 1997.
- [19] F. Ling, D. Manolakis, and J. G. Proakis, "Numerically Robust Least-squares lattice-ladder algorithms with direct updating of the reflection coefficients," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 4, pp. 837–845, 1986.
-

Ricardo Merched obtained his Ph.D. degree from University of California, Los Angeles (UCLA). He became an Assistant Professor at the Department of Electrical and Computer Engineering, the Federal University of Rio de Janeiro, Brazil. His current main interests include fast adaptive filtering algorithms, multirate systems for echo cancellation, and efficient digital signal processing techniques for MIMO equalizer architectures in wireless and wireline communications.

