A LATTICE SHALVI-WEINSTEIN ALGORITHM FOR BLIND EQUALIZATION

Dario B. Bernardes[1], *Maria D. Miranda*[2], *and Magno T. M. Silva*[3]

[1]Pós-Graduação em Engenharia Elétrica, Universidade Presbiteriana Mackenzie,
[2,3]Escola Politécnica, Universidade de São Paulo
Av. Prof. Luciano Gualberto, trav. 3, nº 158, CEP 05508-900 – São Paulo, SP, Brazil
E-mails: dbugatti@gmail.com, maria@lcs.poli.usp.br, magno@lps.usp.br

ABSTRACT

In blind equalization, the Constant Modulus Algorithm (CMA) and Shalvi-Weinstein Algorithm (SWA) present an unfavorable tradeoff between convergence rate and computational cost. Inspired in supervised order-recursive algorithms, we propose a Lattice SWA, which has the number of operations per iteration of the equalizer order and maintains the SWA convergence rate. It presents a more robust behavior than that of SWA, avoiding numerical divergence when implemented in finite precision.

Index Terms— Adaptive filters, blind equalization, order-recursive algorithms, lattice structure, Shalvi-Weinstein Algorithm.

1. INTRODUCTION

Adaptive equalizers are widely used in modern digital communication systems to remove intersymbol interference introduced by dispersive channels. In the supervised equalization, LMS (Least Mean-Square) and RLS (Recursive Least-Squares) are the most employed algorithms [1]. In blind equalization, among the most popular adaptive schemes are the Constant Modulus Algorithm (CMA) [2] and the Shalvi-Weinstein Algorithm (SWA) [3].

It is well-known in the literature that LMS and RLS present an unfavorable tradeoff between convergence rate and computational cost [1]. The order-recursive RLS algorithms, such as Least-Squares Lattice (LSL), can be more adequate, since they have computational complexity of the equalizer order and maintain the RLS convergence rate [1]. An important member of this algorithm family is the Error Feedback-LSL (EF-LSL) [4]. Although no proof of its numerical stability is known, it has always been observed to yield reliable numerical results, even in finite precision [5].

Based on the link between blind equalization and classical adaptive filtering of [6], CMA and SWA can be interpreted as blind versions of LMS and RLS, respectively. Thus, they present equivalent behaviors compared to those of supervised algorithms. As the RLS algorithm, SWA presents instability problems and can diverge in some circumstances, after the initial convergence. In general, this occurs when the forgetting factor is not very close to one. In this context, designing a stable algorithm which has a more favorable tradeoff between convergence rate and computational cost is a problem of interest.

Inspired in the order-recursive RLS algorithms obtained in [7, 5] and in the general methodology for the design of blind adaptive algorithms [6], we propose a Lattice-SWA (L-SWA) based on prediction errors. The proposed algorithm presents a computational complexity of the equalizer order and can avoid situations of divergence, if adequately implemented.

Throughout the paper, we assume real data, without loss of generality. In Section 2, the problem formulation is presented. In Section 3, SWA is obtained from a deterministic cost function. Then, L-SWA is introduced in Section 4. Simulation results, confirming the numerical robustness of the algorithm, and the conclusions are presented in sections 5 and 6, respectively.

2. PROBLEM FORMULATION

A simplified communication system is depicted in Figure 1. The signal $a(n)$, assumed independent, identically distributed, and non Gaussian, is transmitted through an unknown channel, whose model is constituted by an FIR (Finite Impulse Response) filter $H(z)$ and additive white Gaussian noise $\eta(n)$. From the received signal $u(n)$ and the known statistical properties of the transmitted signal, the blind equalizer must mitigate the channel effects and recover the signal $a(n)$ for some delay τ_d. The output of the equalizer is $y(n) = \mathbf{u}^T(n)\mathbf{w}$, where T stands for the transpose of a vector, $\mathbf{u}(n)$ is the input regressor vector, and $\mathbf{w}$ the equalizer weight vector (both with M coefficients).

As the constant modulus [2] and super-exponential [3] blind schemes are equivalent under certain circumstances [8], we consider only the constant modulus cost function,

$$J_G = \mathrm{E}\left\{(y^2(n) - r)^2\right\}, \qquad (1)$$

This work was supported by FAPESP (02/12216-3).

where $r = \mathrm{E}\{a^4(n)\}/\mathrm{E}\{a^2(n)\}$ and $\mathrm{E}\{\cdot\}$ stands for expectation operation. Gradient and quasi-Newton methods were exploited in minimizing J_G, leading to several different algorithms. CMA is based on a stochastic gradient approach [2] and is the most popular due to its simplicity of implementation. However, the theoretical conditions to ensure its convergence and stability remain an open problem. SWA was originally derived in [3] from the Super-Exponential cost function and can be interpreted as a stochastic gradient algorithm with an optimal step-size [8]. It can also be interpreted as a quasi-Newton algorithm if the autocorrelation matrix, responsible for the whitening of the input sequence [3, 1], is assumed to be an approximation of the Hessian matrix of J_G. Its computational complexity is of the order of M^2, which can be inadequate for many practical situations.

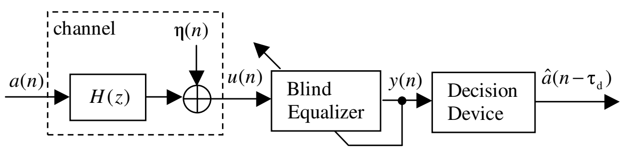

Fig. 1. Schematic representation of a communication system.

3. A DETERMINISTIC COST FUNCTION AND SWA

Inspired in the least-mean-squares criterion and in (1), we consider the cost function

$$J(n) = \sum_{\ell=0}^{n} \lambda^{n-\ell}(y^2(\ell) - r)^2, \quad (2)$$

where $y(\ell) = \mathbf{u}^T(\ell)\mathbf{w}(n)$ and $0 \ll \lambda < 1$. When the gradient of $J(n)$ with respect to $\mathbf{w}(n)$ is a null vector, we obtain

$$r\mathbf{\Phi}(n)\mathbf{w}(n) = \mathbf{\Theta}(n), \quad (3)$$

where

$$\mathbf{\Phi}(n) = \sum_{\ell=0}^{n} \lambda^{n-\ell}\mathbf{u}(\ell)\mathbf{u}^T(\ell)$$

and

$$\mathbf{\Theta}(n) = \sum_{\ell=0}^{n} \lambda^{n-\ell}y^3(\ell)\mathbf{u}(\ell).$$

Assuming that the solution of the instant $(n-1)$ is known, an update for $\mathbf{w}(n)$ can be obtained from (3). Thus, noting that $\mathbf{\Phi}(n) = \lambda\mathbf{\Phi}(n-1) + \mathbf{u}(n)\mathbf{u}^T(n)$, we can rewrite (3) at instant n as

$$r\mathbf{\Phi}(n)\mathbf{\Delta w}(n) - \mathbf{g}(n) = (\bar{y}^2(n) - r)\bar{y}(n)\mathbf{u}(n),$$

where

$$\mathbf{\Delta w}(n) = \mathbf{w}(n) - \mathbf{w}(n-1),$$
$$\bar{y}(\ell) = \mathbf{u}^T(\ell)\mathbf{w}(n-1),$$

and

$$\mathbf{g}(n) = \sum_{\ell=0}^{n} \lambda^{n-\ell}(y^3(\ell) - \bar{y}^3(\ell))\mathbf{u}(\ell).$$

Using the approximation

$$\mathbf{g}(n) \approx 3\mathrm{E}\{a^2(n)\}\mathbf{\Phi}(n)\mathbf{\Delta w}(n), \quad (4)$$

the update of the coefficient vector can be written as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{1}{r - 3\mathrm{E}\{a^2(n)\}}e(n)\mathbf{\Phi}(n)^{-1}\mathbf{u}(n), \quad (5)$$

where

$$e(n) = \bar{y}(n)(\bar{y}^2(n) - r). \quad (6)$$

Eq. (5) characterizes SWA. It is relevant to note that: (i) from a cost function based on the second and fourth order cumulants of the equalizer output, this algorithm was derived in [3], using empirical approximations for cumulants; (ii) different from RLS, which provides an exact solution for least-squares criterion, (5) is not an exact solution for the minimization of (2) due to the approximation (4).

4. LATTICE-SWA

Using a state-space representation, (5) can be rewritten as

$$\begin{bmatrix} \mathbf{w}(n) \\ \gamma(n)(d(n) - \bar{y}(n)) \end{bmatrix} = \mathbf{A}(n)\begin{bmatrix} \mathbf{w}(n-1) \\ d(n) \end{bmatrix}, \quad (7)$$

where $\gamma(n) = 1 - \mathbf{u}^T(n)\mathbf{\Phi}^{-1}(n)\mathbf{u}(n)$ and

$$\mathbf{A}(n) = \begin{bmatrix} \mathbf{I} - \mathbf{\Phi}^{-1}(n)\mathbf{u}(n)\mathbf{u}^T(n) & \mathbf{\Phi}^{-1}(n)\mathbf{u}(n) \\ -\gamma(n)\mathbf{u}^T(n) & \gamma(n) \end{bmatrix}$$

is the state-transition matrix. The scalar

$$d(n) = x(n)\bar{y}(n), \quad (8)$$

with

$$x(n) = \frac{|\bar{y}(n)|^2 - 3\mathrm{E}\{a^2(n)\}}{r - 3\mathrm{E}\{a^2(n)\}}, \quad (9)$$

can be interpreted as an estimate of the desired response. Eq. (7) has the same structure of the RLS state-space representation used in [9]. However, in the RLS case, $d(n)$ does not depend on $\mathbf{w}(n-1)$, and here, $\mathbf{w}(n-1)$ is fed back through a nonlinear mechanism in the evaluation of $d(n)$.

Let $\mathbf{\Phi}(n) = \mathbf{K}^T(n)\mathbf{D}(n)\mathbf{K}(n)$ be the Cholesky factorization [1]. The matrix $\mathbf{K}^T(n)$ is upper triangular with 1's along its main diagonal; all of its elements below the main diagonal are zero. Moreover, each line represents the coefficients of the backward prediction error filter, whose order corresponds to the position of that row in the matrix. $\mathbf{D}(n)$ is a diagonal matrix, whose i^{th} element represents the backward prediction error energy of a filter of order i. Thus,

$$\mathbf{b}(n) = \mathbf{K}^{-T}(n)\mathbf{u}(n) \quad (10)$$

represents the *a posteriori* backward prediction error vector and

$$\boldsymbol{\psi}(n) = \mathbf{K}^{-T}(n-1)\mathbf{u}(n) \tag{11}$$

the *a priori* backward prediction error vector. Defining

$$\mathbf{v}(n) = \mathbf{K}(n)\mathbf{w}(n), \tag{12}$$

from (7), we arrive at

$$\left[\begin{array}{c} \mathbf{v}(n) \\ \gamma(n)(d(n)-\bar{y}(n)) \end{array}\right] = \mathbf{B}(n)\left[\begin{array}{c} \lambda\mathbf{v}(n-1) \\ d(n) \end{array}\right], \tag{13}$$

where

$$\mathbf{B}(n) = \left[\begin{array}{cc} \boldsymbol{\Gamma}^{-1}(n)\boldsymbol{\Gamma}(n-1) & \mathbf{D}^{-1}(n)\mathbf{b}(n) \\ -\lambda^{-1}\gamma(n)\boldsymbol{\psi}^T(n) & \gamma(n) \end{array}\right]$$

and $\boldsymbol{\Gamma}(n) = \mathbf{K}^T(n)\mathbf{D}(n)$.

We can use (13) to implement SWA using a lattice structure. The resulting algorithm is named Lattice-SWA (L-SWA). Each lattice stage provides prediction errors in its output. From these errors, it is possible to obtain an estimate of the desired response [1]. The literature contains different versions of algorithms to obtain prediction errors from the observed sequence $\{u(n)\}$. However, the modified EF-LSL presents reliable numerical properties, even in the absence of persistent excitation and when implemented in finite precision [5].

From the previous observations, L-SWA, summarized in Table 1, uses the modified EF-LSL algorithm of [5] for the prediction section. The variables $(E_i^f(n),\ \eta_i,\ k_i^f(n))$ and $(E_i^b(n),\ \psi_i(n), k_i^b(n))$ represent respectively, energies, *a priori* prediction errors and reflection coefficients of the forward and backward predictions and $\gamma_i(n)$ are the conversion factors. The variables $(b, \bar{b}, f, \bar{f})$ were introduced to reduce the computational complexity of the algorithm. To ensure robust numerical behavior in the prediction section, it is necessary to avoid divisions by values close to zero in their computations. To this end, we add a small positive constant δ to the denominators, whose value depends on the implementation precision.

For the joint estimation section, using (11) and (12), we rewrite $\bar{y}(n) = \mathbf{u}^T(n)\mathbf{w}(n-1)$ as $\bar{y}(n) = \boldsymbol{\psi}^T(n)\mathbf{v}(n-1)$. The estimation errors α_i, $i = 1, 2, \cdots, M-1$ are obtained from the backward prediction errors and the coefficients $v_i(n-1)$. The zero-order estimation error is $\alpha_0 = d(n)$.

In the supervised case, the numerical robustness of the algorithm is ensured through the stability of the prediction section [7]. In the blind case, the desired response $d(n)$ is estimated with $\bar{y}(n)$ multiplied by a correction factor $x(n)$, as shown in (8). Note that $x(n)$ and $\bar{y}(n)$ are obtained from a feedback mechanism, which can cause numerical divergence. Thus, we also need to include conditions to ensure the stability of the joint estimation section. To avoid numerical divergence, the correction factor $x(n)$ must always be positive, which makes that $d(n)$ has the same sign of $\bar{y}(n)$. If $x(n)$ is negative, we make $d(n) = 0$. In this case, the algorithm rejects the estimate of the desired response at the instant n. Using this condition, through exhaustive simulations, we have not observed divergence, even for small values of λ. Moreover, the behavior of the algorithm is not significatively affected, as shown in the next section.

Table 1 summarizes L-SWA. The variables, which are initialized with non null values, are listed in the top of this table. The initialization of $\mathbf{v}$ is the same of $\mathbf{w}$ in SWA, that is, it uses the center-tap initialization method. The proposed algorithm requires a computation of $(14M + 4)$ multiplications, $2M$ divisions and $12M$ additions. As its computational cost is of the equalizer order, L-SWA can be interpreted as a fast version of SWA.

Table 1. Summary of L-SWA.

Initialization:
$v_\Delta(-1) = 1$
$\{E_i^f(0) = E_i^b(-1) = \lambda\}$, $i = 0, \ldots, M-1$

for $n = 1, 2, 3, \ldots$ do:
  $\eta_0 = \psi_0(n) = u(n)$
  $\alpha_0 = d(n-1)$; $\gamma_0 = 1$
  for $i = 0 : M-1$,
    $b = \psi_i(n-1)\gamma_i$
    $f = \eta_i\gamma_i$
    $E_i^b(n-1) = \lambda E_i^b(n-2) + \psi_i(n-1)\ b$
    $E_i^f(n) = \lambda E_i^f(n-1) + \eta_i\ f$
    $\bar{b} = b/(\delta + E_i^b(n-1))$
    $\bar{f} = f/(\delta + E_i^f(n))$
    $\gamma_{i+1} = \gamma_i - \bar{b}\ b$
  Lattice:
    $\psi_{i+1}(n) = \psi_i(n-1) - k_i^b(n-1)\eta_i$
    $\eta_{i+1} = \eta_i - k_i^f(n-1)\psi_i(n-1)$
    $k_i^f(n) = k_i^f(n-1) + \eta_{i+1}\bar{b}$
    $k_i^b(n) = k_i^b(n-1) + \bar{f}\psi_{i+1}(n)$
  Joint estimation:
    $\alpha_{i+1} = \alpha_i - \psi_i(n-1)\ v_i(n-1)$
    $v_i(n-1) = v_i(n-2) + \bar{b}\ \alpha_{i+1}$
  end
  $\bar{y}(n) = \boldsymbol{\psi}^T(n)\mathbf{v}(n-1)$
  $x(n) = (\bar{y}^2(n) - 3\mathrm{E}\{a^2(n)\})/(r - 3\mathrm{E}\{a^2(n)\})$
    if $x(n) < 0$ $x(n) = 0$ end
  $d(n) = x(n)\ \bar{y}(n)$
end

5. SIMULATION RESULTS

In this section, we compare L-SWA of Table 1 to SWA in different situations. We assume binary signals, the channel $H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2}$, and an equalizer with $M = 11$ coefficients.

Figure 2 shows the equalizer output and the squared error defined in (6), considering absence of noise, $\lambda = 0.8$, $h_0 = h_2 = 0.4$ and $h_1 = 1$. We observe that SWA does not work properly in this situation, diverging after 630 iterations. L-SWA converges after 300 iterations, maintaining a stable and adequate behavior. For a channel with less intersymbol interference, i.e., $h_0 = h_2 = 0.1$, SWA and L-SWA present exactly the same behavior for the first 870 iterations, as shown in Figure 3. After that, SWA diverges and L-SWA presents errors during 100 iterations, returning to its adequate behavior. We should notice that these errors are caused by the imposition of $d(n) = 0$. However, this condition is necessary to ensure its stability.

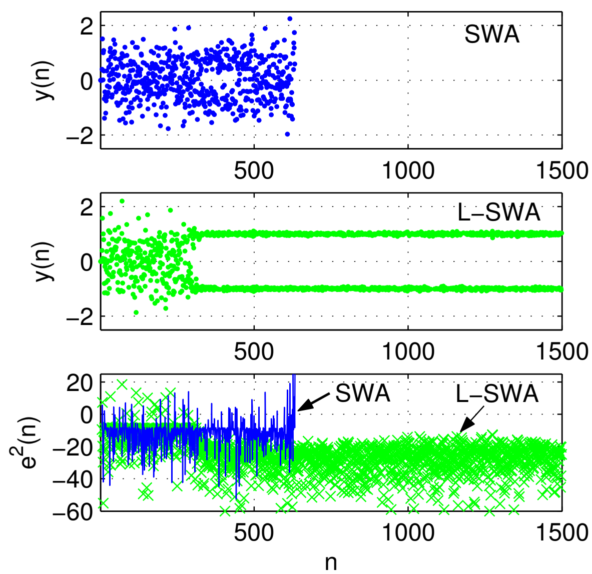

Fig. 2. Equalizer output and $e^2(n)$ in dB with $M = 11$, $\lambda = 0.8$, $h_0 = h_2 = 0.4$, and $h_1 = 1$.

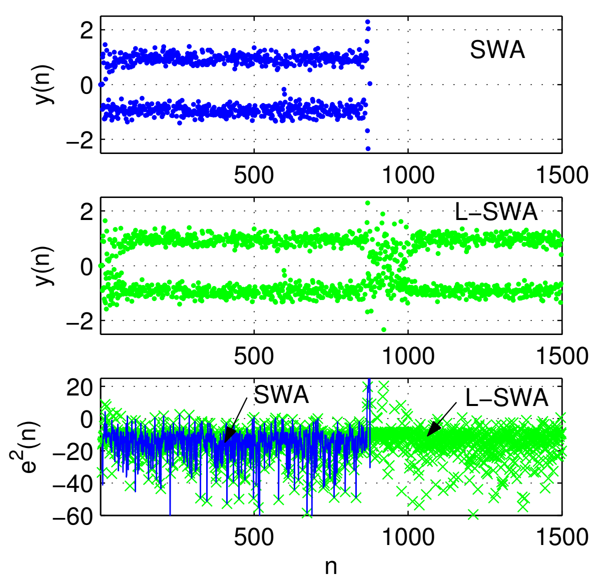

Fig. 3. Equalizer output and $e^2(n)$ in dB with $M = 11$, $\lambda = 0.8$, SNR=20 dB, $h_0 = h_2 = 0.1$, and $h_1 = 1$.

It general, when λ is close to one, SWA does not diverge and both algorithms present exactly the same performance. The advantages of L-SWA is its lower computational complexity, and an adequate behavior for smaller forgetting factors.

6. CONCLUSIONS

We proposed a blind Lattice Shalvi-Weisntein Algorithm, which avoids numerical divergence, has a computational complexity of the equalizer order, and maintains the SWA converge rate. Although there is no proof of its numerical stability, the simulations suggest that it presents a robust behavior, even for small forgetting factors. Further work should imply a multiple-input multiple-output version for space-time equalization.

7. REFERENCES

[1] S. Haykin, *Adptive Filter Theory*, Prentice Hall, 3rd edition, Upper Saddle River, 1996.

[2] D. N. Godard, "Self-recovering equalization and carrier tracking in two dimensional data communication system," *IEEE Trans. on Commun.*, vol. 28, pp. 1867–1875, Nov. 1980.

[3] O. Shalvi and E. Weinstein, "Super-exponential methods for blind deconvolution," *IEEE Trans. on Inform. Theory*, vol. 39, pp. 504–519, Mar. 1993.

[4] F. Ling, D. Monolakis, and J. G. Proakis, "Numerically robust least-squares lattice-ladder algorithms with direct updating of the reflection coefficients," *IEEE Trans. on Acoutics, Speech and Signal Process.*, vol. ASSP-34, pp. 837–845, Aug. 1986.

[5] M. D. Miranda, M. Gerken, and M. T. M. Silva, "Efficient implementation of the a priori error-feedback LSL algorithm," *IEE Eletronics Letters*, vol. 35, pp. 1308–1309, Aug. 1999.

[6] C. B. Papadias and D. T. M. Slock, "Normalized sliding window constant modulus and decision-direct algorithms: a link between blind equalization and classical adaptive filtering," *IEEE Trans. on Signal Process.*, vol. 45, pp. 231–235, Jan. 1997.

[7] P. A. Regalia, "Numerical stability properties of a qr-based fast least squares algorithm," *IEEE Trans. on Signal Process.*, vol. 41, pp. 2096–2109, June 1993.

[8] M. Mboup and P. A. Regalia, "A gradient search interpretation of the super-exponential algorithm," *IEEE Trans. on Inform. Theory*, vol. 46, pp. 2731–2734, Nov. 2000.

[9] P. A. Regalia and M. G. Bellanger, "On the duality between fast QR methods and lattice methods in least-squares adaptive filtering," *IEEE Trans. on Signal Process.*, vol. 39, pp. 879–891, Apr. 1991.