LOW COST ADAPTIVE ALGORITHM FOR BLIND CHANNEL IDENTIFICATION AND SYMBOL ESTIMATION

EUSIPCO (Finland), pp. 1549-1552, Sept. 2000.

Florence Alberge, Pierre Duhamel, Mila Nikolova
ENST/TSI, 46, rue Barrault, 75634 Paris Cedex 13, FRANCE
Phone: (33) 01 45 81 73 65 - Fax (33) 01 45 88 79 35
e-mail :{alberge,duhamel,nikolova}@tsi.enst.fr

ABSTRACT

A recursive/adaptive least squares algorithm (MLRA) is proposed to solve the joint blind channel identification and blind symbol estimation problem. It is based on Deterministic Maximum Likelihood methods (DML) which are pertinent in this field. We prove that when the MLRA converges then it converges towards the global minimum. The MLRA is able to track variations of the system by the introduction of an exponential forgetting factor in the DML criterion. The link between the adaptive algorithm and a Soft Decision Feedback Equalizer is emphasized. Update strategies of the filters can be either of a least squares type or of a stochastic gradient type. Both of them are derived in the paper. Numerical simulations show that the simplifications involved result in small degradations on the performances.

1 Introduction

Blind identification is an important problem in many areas and especially in wireless communications. Our work addresses Single Input Multiple Outputs (SIMO) systems.The SIMO equalization problem can be solved using second-order statistics only, as long as the sub-channels do not share any common zeros. In a fast fading environment, the data relevant to each channel are too few to build reliable statistical estimates. In that case, the problem may be solved by treating the input as a deterministic variable. This paper focuses on this situation. More precisely, this paper deals with Deterministic Maximum Likelihood (DML) methods. ML methods are appealing since, asymptotically, the variance of the estimator achieves the Cramer-Rao lower bound. Among the major contributions to DML methods, we can cite the Two Steps Maximum Likelihood (TSML) [1] and the Iterative Quadratic Maximum Likelihood (IQML) [2]. These algorithms have good convergence rate but their computational cost is too high for working with large data sequences. One way of overcoming this drawback is usually to derive recursive algorithms, which is quite complicated based on this class of algorithms. Another DML based method is the MLBA (Maximum Likelihood Block Algorithm) which was proposed by the authors in [3]. Like the TSML or the IQML, the MLBA is not suited for practical applications with large data sequences. However the structure of the latter is well adapted to derive easily a low cost recursive algorithm.

In this paper, we first recall several properties of the MLBA [4, 3]. We lay particular stress on a test based on the stability/instability of the global/local minimum over a recursive procedure. Then, we derive a recursive version of the MLBA. We prove that when the recursive algorithm converges then it converges towards the global minimum. System adaptivity is then obtained by introducing an exponential weighting factor in the criterion. The connection between this algorithm and the structure of the Decision Feedback Equalizer is emphasized. A low cost version of the recursive algorithm is also presented.

2 Problem formulation

Consider a sequence of unknown binary symbols $\{\widetilde{s}(k)\}$ transmitted through L unknown channels $\widetilde{h}_i$, $1 \leq i \leq L$. Let $x_i(k)$ denote the output of the i^{th} channel at time k, and $\mathbf{X}_N(n) = [x_1(n)...x_L(n)....x_1(n-N+1)...x_L(n-N+1)]^T$ the LN-length vector obtained by interleaving the output of the various channels. This output is modeled as:

$$\mathbf{X}_N(n) = \mathcal{T}_N(\widetilde{\mathbf{h}})\widetilde{\mathbf{s}}_N(n) + \mathbf{B}_N(n) \qquad (1)$$

where $\widetilde{\mathbf{h}}(k) = [\widetilde{h}_1(k)...\widetilde{h}_L(k)]^T$ are the different channels and $\mathbf{B}_N(n)$ stands for the noise vector. The noise sequence on each sensor is assumed to be $i.i.d.$ and the different sequences are mutually uncorrelated. In (1), the operator $\mathcal{T}_N$ transforms a sequence of channels $\mathbf{h}(k) = [h_1(k)...h_L(k)]^T$ into the following LN×(M+N) generalized Sylvester matrix:

$$\mathcal{T}_N(\mathbf{h}) = \begin{pmatrix} \mathbf{h}(0) & \dots & \mathbf{h}(M) & & \\ & \ddots & \dots & \ddots & \\ & & \mathbf{h}(0) & \dots & \mathbf{h}(M) \end{pmatrix}$$

Below, $\mathbf{s}_N(n) = [s(n)...s(n-N-M+1)]^T$ denotes any vector of $M+N$ symbols, where M is the maximum order of a channel. Let us introduce operator $\mathcal{U}$ which transforms a vector $\mathbf{s}_N(n)$ into a LN×L(M+1) matrix, $\mathcal{U}(\mathbf{s}_N(n))$, in such a way that [3]:

$$\mathcal{U}(\mathbf{s}_N(n))\mathbf{h} = \mathcal{T}_N(\mathbf{h})\mathbf{s}_N(n), \ \forall \mathbf{s}_N, \ \forall \mathbf{h} \qquad (2)$$

In the following, we assume that both matrices $\mathcal{T}_N(\widetilde{\mathbf{h}})$ and $\mathcal{U}(\widetilde{\mathbf{s}}_N)$ are full column rank, that M is known and that $\{\widetilde{s}(k)\}$ has linear complexity $2M+1$ or greater [5].

3 A maximum likelihood block algorithm

The work presented here is based on the MLBA (Maximum Likelihood **Block** Algorithm). So, the main results [6, 3, 4] about the latter are briefly recalled in this section. We consider the minimization of the following criterion:

$$\mathcal{J}(\mathbf{h}, \mathbf{s}_N(n)) = \|\mathbf{X}(n) - \mathcal{T}_N(\mathbf{h})\mathbf{s}_N(n)\|^2$$

Using (2), $\mathcal{J}$ can also be rewritten as:

$$\mathcal{J}(\mathbf{h}, \mathbf{s}_N(n)) = \|\mathbf{X}_N(n) - \mathcal{U}(\mathbf{s}_N(n))\mathbf{h}\|^2$$

The MLBA is obtained thanks to the dual expressions of the criterion. After some initialization, one iterates the following two steps until convergence:

$$\widehat{\mathbf{h}}^{(k)} = [\mathcal{U}(\widehat{\mathbf{s}}_N^{(k-1)})^H \mathcal{U}(\widehat{\mathbf{s}}_N^{(k-1)})]^{-1} \mathcal{U}(\widehat{\mathbf{s}}_N^{(k-1)})^H \mathbf{X}_N(n) \quad (3)$$

$$\widehat{\mathbf{s}}_N^{(k)} = [\mathcal{T}_N(\widehat{\mathbf{h}}^{(k)})^H \mathcal{T}_N(\widehat{\mathbf{h}}^{(k)})]^{-1} \mathcal{T}_N(\widehat{\mathbf{h}}^{(k)})^H \mathbf{X}_N(n) \quad (4)$$

where $\mathcal{T}_N(\widehat{\mathbf{h}}^{(k)})$ and $\mathcal{U}(\widehat{\mathbf{s}}_N^{(k)})$ are assumed to be full column rank. The following properties hold for the MLBA [4, 3]:

(P1) CONVERGENCE: Each step diminishes the value of $\mathcal{J}$ and the MLBA converges possibly to a local minimum.

(P2) UNIQUENESS OF THE GLOBAL MINIMUM: **Theorem 1**: *If* $\mathbf{B}_n(n) = \mathbf{0}_{LN}$, *if* $\mathcal{T}_N(\widetilde{\mathbf{h}})$ *is full column rank and if* $\{\widetilde{s}(k)\}$ *has linear complexity* $2M+1$ *or greater then* $\mathcal{J}(\widehat{\mathbf{h}}, \widehat{\mathbf{s}}_N(n)) = 0$ *iff* $\exists\ \alpha$ *such as* $\widehat{\mathbf{h}} = \alpha\widetilde{\mathbf{h}}$ *and* $\widehat{\mathbf{s}}_N(n) = \widetilde{\mathbf{s}}_N(n)/\alpha$ (see [4] for the proof).

(P3) STABILITY OF THE ESTIMATES IN A RECURSIVE PROCEDURE: The recursive procedure BGWT (Block Growing Window Technique) is composed of the following steps:

- **Step 0** : Minimize $\mathcal{J}(\mathbf{h}, \mathbf{s}_N)$ using the MLBA to obtain $(\widehat{\mathbf{h}}^{(0)}, \widehat{\mathbf{s}}_N^{(0)}(n))$.
- **Step** $k = 1, \ldots, K-1$, **where** $K = 3M+1$: A new symbol $\widetilde{s}(n+k)$ is transmitted. Then, the considered criterion $\mathcal{J}(\mathbf{h}, \mathbf{s}_{N+k})$ is defined over a block of size $N+k$. This is the same criterion as previously except for the size of the block. The minimizer of $\mathcal{J}(\mathbf{h}, \mathbf{s}_{N+k})$ is $(\widehat{\mathbf{h}}^{(k)}, \widehat{\mathbf{s}}_{N+k}^{(k)}(n+k))$. Step k is initialized with $\widehat{\mathbf{h}}^{(k-1)}$.

At the end of these iterations, either the channel estimates remain unchanged *i.e.* $\widehat{\mathbf{h}}^{(k)} = \widehat{\mathbf{h}}\ \forall k$, or not ($\exists\ i, k \in [1, .., K]$ such as $\widehat{\mathbf{h}}^{(k)} \neq \widehat{\mathbf{h}}^{(i)}$). The consequences of these issues are formalized below.

Theorem 2: *Assume that* $\mathbf{B}_{N+K}(n+K) = \mathbf{0}_{L(N+K)}$ *and that the channel* $\widetilde{\mathbf{h}}$ *is constant over the window* $[n-N-M+1, \ldots, n+3M+1]$. *Assume also that* $\mathcal{T}_{N+k}(\widehat{\mathbf{h}})$, $\mathcal{T}_{N+k}(\widetilde{\mathbf{h}})$, $\mathcal{U}(\widehat{\mathbf{s}}_{N+k}(n+k))$, $\mathcal{U}(\widetilde{\mathbf{s}}_{N+k}(n+k))$ *are full column rank* $\forall k$ *and that* $\{\widetilde{s}(k)\}$ *has linear complexity* $2M+1$ *or greater.*
If all along the procedure BGWT, we have $\widehat{\mathbf{h}}^{(k)} = \widehat{\mathbf{h}}$ *and* $\widehat{\mathbf{s}}_{N+k}^{(k)}(n+k) = \widehat{\mathbf{s}}_{N+k}(n+k) \neq \mathbf{0}_{M+N+k}$ *for any* $k = 0, \ldots, K-1$ *then* $(\widehat{\mathbf{h}}, \widehat{\mathbf{s}}_{N+K}(n+K))$ *is the global minimum of* $\mathcal{J}(\mathbf{h}, \mathbf{s}_{N+K})$ *up to a scalar factor.* (see [3] for the proof)

Theorem 2 proves that the global minimum is the only stable point for $K = 3M+1$ consecutive steps. This remark justifies our choice to develop recursive algorithms. The procedure BGWT is not practical for a great number of data. In the next section, we propose a lower cost algorithm.

4 Maximum Likelihood Recursive Algorithm

4.1 Derivation of the recursive algorithm

Now, we derive a recursive algorithm where the updated estimates of the symbols at iteration i are calculated based on both their least-squares estimates at iteration $i-1$ and a newly arrived datum. The MLRA (ML **Recursive** Algorithm) follows from the following simplifications :

(S1) The iterative minimization w.r.t. the joint variable in BGWT is replaced by a minimization w.r.t. each variable separately. So, at step i, we calculate:

$$\widehat{\mathbf{s}}_{N+i}^{(i)}(n+i) = \arg\min_{\mathbf{s}_{N+i}} \mathcal{J}(\widehat{\mathbf{h}}^{(i-1)}, \mathbf{s}_{N+i}) \quad (5)$$

$$\widehat{\mathbf{h}}^{(i)} = \arg\min_{\mathbf{h}} \mathcal{J}(\mathbf{h}, \widehat{\mathbf{s}}_{N+i}^{(i)}(n+i)) \quad (6)$$

These relations coincide with the first iteration of BGWT.

(S2) At iteration i, the BGWT procedure computes $N+i$ symbols. Hence, the computational complexity of equation (5) increases quickly with i. To avoid that problem, and considering that it is unlikely that the most recent received samples have a strong impact on the estimate of the symbols that have been emitted long ago, we update only the last P symbols. Hence, P is a fundamental parameter to be determined, which will drive a complexity/efficiency trade-off. Implicitly, the others symbols are thus supposed to be correctly estimated. The minimization w.r.t. the symbols reduces to:

$$\widehat{\mathbf{s}}_{P+1-M}^{(i)}(n+i) = \arg\min_{\mathbf{Z}\in\mathbb{C}^{P+1}} \|\mathbf{X}_{P+1}(n+i) - \mathcal{T}_{P+1}(\widehat{\mathbf{h}}^{(i-1)}) \begin{bmatrix} \mathbf{Z} \\ \widehat{\mathbf{s}}_0^{(i-1)}(n+i-P-1) \end{bmatrix} \|^2$$

(S3) The estimated channel $\widehat{\mathbf{h}}^{(i)}$ is updated recursively from $\widehat{\mathbf{h}}^{(i-1)}$ which is done without any approximation.

Finally, the estimated symbols at iteration i read:

$$\widehat{\mathbf{s}}_{P+1-M}^{(i)}(n+i) = \mathcal{A}_{P+1}^{\sharp}(\widehat{\mathbf{h}}^{(i-1)}) \left[\mathbf{X}_{P+1}(n+i) - \mathcal{B}_{P+1}(\widehat{\mathbf{h}}^{(i-1)}) \widehat{\mathbf{s}}_0^{(i-1)}(n+i-P-1)\right] \quad (7)$$

where the superscript $\sharp$ denotes Moore-Penrose pseudo-inverse and $\mathcal{A}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})$ and $\mathcal{B}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})$ are the sub-matrices of $\mathcal{T}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})$ defined as:

$$\mathcal{T}_{P+1}(\widehat{\mathbf{h}}^{(i-1)}) = [\ \underbrace{\mathcal{A}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})}_{P+1}\ \vdots\ \underbrace{\mathcal{B}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})}_{M}\ ] \quad (8)$$

The filter $\widehat{\mathbf{h}}^{(i)}$ is updated thanks to the equation below:

$$\widehat{\mathbf{h}}^{(i)} = \widehat{\mathbf{h}}^{(i-1)} + \left[\mathbf{R}_R^{(i)}\right]^{-1} \Big\{ \mathcal{U}^H(\widehat{\mathbf{s}}_{P+1}^{(i)}(n+i)) \left[\mathbf{X}_{P+1}(n+i) - \mathcal{U}(\widehat{\mathbf{s}}_{P+1}^{(i)}(n+i))\widehat{\mathbf{h}}^{(i-1)}\right] - \mathcal{U}^H(\widehat{\mathbf{s}}_P^{(i-1)}(n+i-1)) \times \left[\mathbf{X}_P(n+i-1) - \mathcal{U}(\widehat{\mathbf{s}}_P^{(i-1)}(n+i-1))\widehat{\mathbf{h}}^{(i-1)}\right] \Big\} \quad (9)$$

where $\mathbf{R}_R^{(i)} = \mathcal{U}(\widehat{\mathbf{s}}_{N+i}^{(i)}(n+i))^H \mathcal{U}(\widehat{\mathbf{s}}_{N+i}^{(i)}(n+i))$. $\mathbf{R}_R^{(i)}$ is computed from $\mathbf{R}_R^{(i-1)}$ using:

$$\mathbf{R}_R^{(i)} = \mathbf{R}_R^{(i-1)} + \mathcal{U}(\widehat{\mathbf{s}}_{P+1}^{(i)}(n+i))^H \mathcal{U}(\widehat{\mathbf{s}}_{P+1}^{(i)}(n+i)) - \mathcal{U}(\widehat{\mathbf{s}}_P^{(i-1)}(n+i-1))^H \mathcal{U}(\widehat{\mathbf{s}}_P^{(i-1)}(n+i-1))$$

The recursive update of $[\mathbf{R}_R^{(i)}]^{-1}$ is then obtained by applying twice the matrix inversion lemma. Eq. (9) combined with the recursive update of $[\mathbf{R}_R^{(i)}]^{-1}$ form the MLRA. A fast calculation scheme for eq. (9) can be obtained similarly to the fast RLS. We leave this point for future work. The following convergence result holds for the MLRA:

Theorem 3: *If the assumptions of theorem 1 are met and if the MLRA converges then, it converges towards the global minimum.*

The proof of theorem 3 has been relegated to section A.

4.2 Initialization

The MLRA, like the TSML [1] and the IQML [2], needs to start from a reliable initialization point. Here, we propose to initialize the MLRA with $(\widehat{\mathbf{h}}^{(0)}, \widehat{\mathbf{s}}_N^{(0)}(n))$ defined as the stationary point of the MLBA (cf. section 3). The choice of N reflects a tradeoff between the accuracy of the estimates and the involved computational cost. Experience show that choosing N about $10M$ leads to a reasonable compromise.

5 Adaptive algorithm

The algorithm must be able to track the variations in a non-stationary environment. So, we introduce a weighting factor in $\mathcal{J}$ to ensure that data in the distant past are forgotten. The MLAA (ML **Adaptive** Algorithm) is derived from the MLRA by replacing $\mathcal{J}$ with $\mathcal{J}_\lambda$ defined as:

$$\mathcal{J}_\lambda(\mathbf{h}, \mathbf{s}_{N+i}) = \sum_{t=n-N+1}^{n+i} \lambda^{n+i-t} \|\mathbf{X}_1(t) - \mathcal{T}_1(\mathbf{h})\mathbf{s}_1\|^2$$

where $\lambda \in [0;1]$ is a forgetting factor. Using a matrix formulation, $\mathcal{J}_\lambda(\mathbf{h}, \mathbf{s}_{N+i})$ reads:

$$\mathcal{J}_\lambda(\mathbf{h}, \mathbf{s}_{N+i}) = \|\Lambda_{N+i}^{1/2}\left[\mathbf{X}_{N+i}(n+i) - \mathcal{T}_{N+i}(\mathbf{h})\mathbf{s}_{N+i}\right]\|^2$$

where $\Lambda_{N+i} = diag([\underbrace{1...1}_{L} \vdots \underbrace{\lambda...\lambda}_{L} \vdots ..... \vdots \underbrace{\lambda^{N+i-1}...\lambda^{N+i-1}}_{L}])$. We replace $\mathcal{J}$ with $\mathcal{J}_\lambda$ in paragraph 4.1 and the MLAA is obtained in the same way as the MLRA. Then, the estimated symbols at iteration i read:

$$\begin{aligned}\widehat{\mathbf{s}}_{P+1-M}^{(i)}(n+i) = \mathcal{C}_{P+1}^{\sharp}(\widehat{\mathbf{h}}^{(i-1)}) \Big[\Lambda_{P+1}^{1/2}\mathbf{X}_{P+1}(n+i) \\ -\mathcal{D}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})\widehat{\mathbf{s}}_0^{(i-1)}(n+i-P-1)\Big] \quad (10)\end{aligned}$$

The filter $\widehat{\mathbf{h}}^{(i)}$ is updated thanks to the equation below:

$$\begin{aligned}\widehat{\mathbf{h}}^{(i)} = \widehat{\mathbf{h}}^{(i-1)} + \left[\mathbf{R}_A^{(i)}\right]^{-1} \Big\{\mathcal{U}^H(\widehat{\mathbf{s}}_{P+1}^{(i)}(n+i))\Lambda_{P+1}\left[\mathbf{X}_{P+1}(n+i)\right. \\ \left.-\lambda\mathcal{U}(\widehat{\mathbf{s}}_{P+1}^{(i)}(n+i))\widehat{\mathbf{h}}^{(i-1)}\Lambda_{P+1}\right] - \mathcal{U}^H(\widehat{\mathbf{s}}_P^{(i-1)}(n+i-1)) \\ \times\left[\mathbf{X}_P(n+i-1) - \mathcal{U}(\widehat{\mathbf{s}}_P^{(i-1)}(n+i-1))\widehat{\mathbf{h}}^{(i-1)}\right]\Big\} \quad (11)\end{aligned}$$

where $\mathbf{R}_A^{(i)} = \mathcal{U}(\widehat{\mathbf{s}}_{N+i}^{(i)}(n+i))^H \Lambda_{P+1}^{1/2} \mathcal{U}(\widehat{\mathbf{s}}_{N+i}^{(i)}(n+i))$ and $\mathcal{C}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})$ and $\mathcal{D}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})$ are the sub-matrices of $\mathcal{T}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})$ defined as:

$$\Lambda_{P+1}^{1/2}\mathcal{T}_{P+1}(\widehat{\mathbf{h}}^{(i-1)}) = [\ \underbrace{\mathcal{C}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})}_{P+1} \ \vdots \ \underbrace{\mathcal{D}_{P+1}(\widehat{\mathbf{h}}^{(i-1)})}_{M}\ ] \quad (12)$$

We can notice that the MLAA is closely related to a DFE. This question is explored in the next section.

6 Link with a Soft Decision Feedback Equalizer

In [6], Gesbert has proposed the Channel Symbol Algorithm (CSA), based on least squares techniques which aims at minimizing $\mathcal{J}_\lambda$. The channels and symbols are updated alternately. For each iteration, we have:

$$\begin{aligned}\bar{s}(n+i) = \arg\min_{z\in\mathbb{C}} \|\mathbf{X}_1(n+i) \\ -\mathcal{T}_1(\widehat{\mathbf{h}}^{(i-1)})\begin{bmatrix} z \\ \widehat{\mathbf{s}}_0(n+i-1)\end{bmatrix}\|^2 \quad (13)\end{aligned}$$

$$\widehat{s}(n+i) = g(\bar{s}(n+i)) \quad (14)$$

$\widehat{\mathbf{h}}^{(i)}$ *updated via RLS*

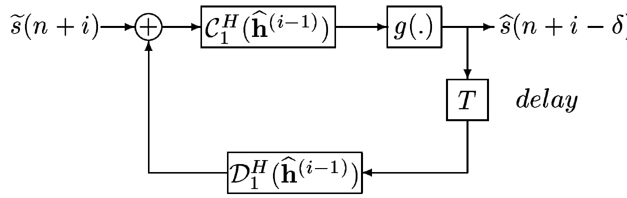

Figure 1: Update of the symbols in the CS algorithm

The operator $g(.)$ stands for a decision device. The link between the CSA and the DFE structure is emphasized in [6] and is summarized below. Using the notations in (12), an explicit expression for $\bar{s}(n+i)$ is obtained:

$$\bar{s}(n+i) = \frac{\mathcal{C}_1^H(\widehat{\mathbf{h}}^{(i-1)})\left(\mathbf{X}_1(n+i) - \mathcal{D}_1(\widehat{\mathbf{h}}^{(i-1)})\widehat{\mathbf{s}}_0(n+i-1)\right)}{\mathcal{C}_1^H(\widehat{\mathbf{h}}^{(i-1)})\mathcal{C}_1(\widehat{\mathbf{h}}^{(i-1)})} \quad (15)$$

The decision feedback structure of eq. (15) and (14) is shown in fig. (1). The main difference between the structure of the CSA and of the DFE is : the presence of a feed-forward filter in the DFE and the presence, in the CSA, of a "spatial" filter $A_1^H(\widehat{\mathbf{h}}^{(i-1)})$ which combines the signals before the decision. Gesbert has also underlined the similarities between our criterion $\mathcal{J}_\lambda(\mathbf{h}, \mathbf{s}_{N+i})$ and a decision directed criterion $\mathcal{J}_{\mathcal{DD}}$ [7] defined as: $\mathcal{J}_{\mathcal{DD}} = \sum_{t=n-N+1}^{n+i} \lambda^{n+i-t}\|\widehat{s}(t) - \bar{s}(t)\|^2$ Replacing (15) in the previous relation, we obtain:

$$\mathcal{J}_{\mathcal{DD}} = \sum_{t=n-N+1}^{n+i} \lambda^{n+i-t} \left\| \frac{\mathcal{C}_1^H(\widehat{\mathbf{h}}^{(i-1)})}{\|\mathcal{C}_1(\widehat{\mathbf{h}}^{(i-1)})\|^2}\left(\mathbf{X}_1(t) - \mathcal{T}_1(\widehat{\mathbf{h}}^{(i-1)})\widehat{\mathbf{s}}_1(t)\right)\right\|^2$$

The difference between the criterion $\mathcal{J}_{\mathcal{DD}}$ and $\mathcal{J}_\lambda$ lies only in the presence of the term $\frac{\mathcal{C}_1^H(\widehat{\mathbf{h}}^{(i-1)})}{\|\mathcal{C}_1(\widehat{\mathbf{h}}^{(i-1)})\|^2}$.
The CSA computes, at each iteration, one and only one symbol whereas the MLAA updates the $P+1$ first symbols in the delay line (cf. fig. (2)). Therefore, possible errors made during the first estimation can be corrected and the error propagation phenomenon frequently observed in the DFE is contained. The absence of cell of decision in the MLAA permits to preserve a linear estimation of the data.

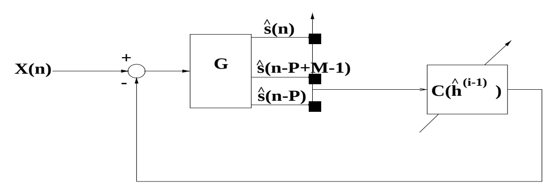

Figure 2: Update of the symbols in the MLAA

7 Further simplifications

In this section, we simplify eq. (9) in order to reduce the computational cost of the MLRA. By construction, $\mathbf{R}_R^{(i)} = diag[\mathcal{S}_l \ ...... \ \mathcal{S}_l]$ and each block $\mathcal{S}_l$ is a matrix filled with symbols estimated at iteration i. We assume that the sequence $\{\widehat{s}^{(i)}(k)\}$ is ergodic then, $lim_{i\to\infty}\frac{1}{N+i}\sum_{l=1-N}^{i}\mathcal{S}_l = \mathbf{Cov}_{M+1}(\widehat{\mathbf{s}}^{(i)})$ where $\mathbf{Cov}_{M+1}(\widehat{\mathbf{s}}^{(i)})$ is the covariance matrix of $\{\widehat{s}^{(i)}\}$. We proved in section 4.1 that, for i after the convergence, $\widehat{\mathbf{s}}_{N+i}^{(i)}(n+i) = \widetilde{\mathbf{s}}_{N+i}(n+i)/\alpha$, $\alpha \in \mathbb{C}^*$, which leads to $\mathbf{Cov}_{M+1}(\widehat{\mathbf{s}}^{(i)}) = \frac{1}{\|\alpha\|^2}\mathbf{Cov}_{M+1}(\widetilde{\mathbf{s}})$. It is generally assumed that $\{\widetilde{s}(k)\}$ is a sequence of $i.i.d.$, complex, circular, random variables with zero mean:

$E[\tilde{s}(k)] = 0 \quad E[\tilde{s}(k)^*\tilde{s}(j)] = \sigma_s^2\delta(k-j) \quad E[\tilde{s}(k)\tilde{s}(j)] = 0$

For i large enough $\mathbf{Cov}_{M+1}(\widehat{\mathbf{s}}^{(i)}) \approx \frac{\sigma_s^2}{\|\alpha\|^2}\mathbf{I}_{M+1}$ and $\mathbf{R}_R^{(i)} \approx \frac{\sigma_s^2(N+i)}{\|\alpha\|^2}\mathbf{I}_{L(M+1)}$. So, eq. (9) is now a stochastic gradient based method with decreasing stepsize.

In the case of the MLAA, we cannot evaluate $\lim_{i\to\infty}\mathbf{R}_A^{(i)}$. The consequence is that we cannot prove that the MLAA is equivalent to a stochastic gradient method. However, simulations show that replacing (11) by a stochastic gradient with constant stepsize result in almost no degradations on the performances. This point is not addressed in this paper, but the corresponding results will be reported in [8].

8 Simulations

We first investigate the influence of the parameter P in the MLRA then, we compare the performances of the MLRA, the MLAA and the BGWT procedure. We present the results on the symbols. For both simulations $\{\tilde{s}(k)\}$ is a QPSK modulation and $\tilde{\mathbf{h}}$ is a mixed-phase channel with $L = 3$ subchannels each of order $M = 3$. In fig. (3), the MSE on the symbols against the SNR is plotted for different values of P. The MSE is averaged over 50 independent noise realizations. The lines corresponding to $P = 1$ and $P = 2$ are incomplete because, at low SNR, the MLRA diverges for some runs. It is clearly seen that choosing $P > 3$ do not really improve the performances. Here the channel order is $M = 3$. Therefore the update of the filter depends only on the symbols in the delay line. In fig. (4), the SNR is set to 10dB and all the algorithms are initialized with $(\widehat{\mathbf{h}}^{(0)}, \widehat{\mathbf{s}}_N^{(0)})$ obtained from the MLBA with $N = 50$. The degradation on the symbol estimation for the MLAA and the MLRA compared to the BGWT procedure remains small (< 2 dB).

9 Conclusion

In this paper, a recursive algorithm (MLRA) based on least-squares techniques is derived. The MLRA follows from various approximations applied to the BGWT procedure in order to diminish the computational complexity. Simulations illustrate that the resulting degradation on the estimates remains reasonably small. Moreover, we proved that when this algorithm converges then it converges towards the global minimum. We can remark that both algorithms are strongly connected with a RLS. So, fast versions could be obtained using techniques similar as for a fast RLS. Furthermore, the update of the filters in the MLRA and the MLAA can be simplified by stochastic gradient techniques. Derivation of the algorithms is straightforward.

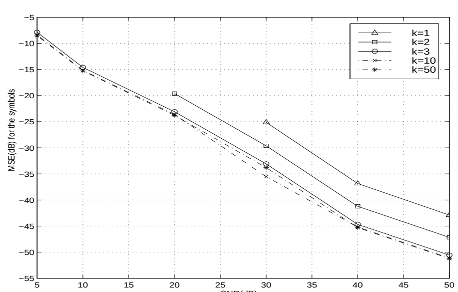

Figure 3: MSE for the symbols vs SNR for the MLRA

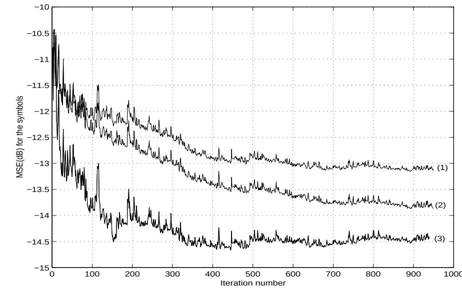

Figure 4: MSE for the symbols vs the iteration number – (1) MLRA P=10, (2) MLAA P=10 and $\lambda = 1 - \frac{1}{6L(M+1)}$, (3) BGWT proc.

A Proof of theorem 3

If the MLRA converges then, $\exists\ n_0$ such as $\forall k \geq n_0\ \widehat{\mathbf{h}}^{(k)} = \widehat{\mathbf{h}}^{(n_0)}$ and $\widehat{\mathbf{s}}_{N+k}^{(k)}(n+k) = \widehat{\mathbf{s}}_{N+k}$. At iteration k, the estimated filter satisfy the following relation:

$$\mathcal{U}(\widehat{\mathbf{s}}_{N+k}^{(k)}(n+k))^H\mathcal{U}(\widehat{\mathbf{s}}_{N+k}^{(k)}(n+k))\widehat{\mathbf{h}}^{(k)} = \mathcal{U}(\widehat{\mathbf{s}}_{N+k}^{(k)}(n+k))^H\mathbf{X}_{N+k}(n+k) \quad (16)$$

Using the fact that eq. (16) is also verified at iteration k-1, eq. (16) reduces to:

$$\mathcal{U}(\widehat{\mathbf{s}}_1(n+k))^H\left[\mathcal{U}(\widehat{\mathbf{s}}_1(n+k))\widehat{\mathbf{h}}^{(n_0)} - \mathbf{X}_1(n+k)\right] = 0 \quad (17)$$

Matrix $\mathcal{U}(\widehat{\mathbf{s}}_1^{(k)}(n+k))^H$ is full column rank as long as $\mathbf{s}_1^{(k)}(n+k) \neq \mathbf{0}_{M+1}$ then the equation above is equivalent to:

$$\mathcal{U}(\widehat{\mathbf{s}}_1(n+k))\widehat{\mathbf{h}}^{(n_0)} - \mathbf{X}_1(n+k) = 0 \quad (18)$$

Eq. (18) holds $\forall\ \ k \geq n_0$. Then, we stack the equations obtained for k, k+1,...k+K with $K \geq 3M+1$ and we get:

$$\mathcal{U}(\widehat{\mathbf{s}}_K)\widehat{\mathbf{h}}^{(n_0)} - \mathbf{X}_K(n+K) = 0$$

which is equivalent to $\mathcal{J}(\mathbf{h}^{(n_0)}, \widehat{\mathbf{s}}_K) = 0$. Since the conditions of theorem 1 are satisfied, then $\widehat{\mathbf{h}}^{(n_0)}$ and $\widehat{\mathbf{s}}_K$ are the true values of the parameters up to a scalar factor.

References

[1] Y. Hua. Fast maximum likelihood for blind identification of multiple FIR channel. *IEEE Trans. on Signal Processing*, 44:661–672, Mar. 1996.

[2] D. Slock. Blind fractionnaly-spaced equalization, perfect reconstruction filterbanks, and multilinear prediction. In *ICASSP Proc*, 1994.

[3] F. Alberge, P. Duhamel, and M. Nikolova. Blind Identification/Equalization using deterministic maximum likelihood and a partial information on the input. In *SPAWC Proc., Annapolis, USA*, 1999.

[4] E. Pite and P. Duhamel. Bilinear methods for blind channel equalization : (no) local minimum issue. In *ICASSP Proc.*, May 1998.

[5] R.E. Blahut. *Algebraic methods for signal procesing and communication coding.* Springer Verlag, NY, 1992.

[6] D. Gesbert, P. Duhamel, and Sylvie Mayrargue. Blind least-square approaches for joint data/channel estimation. In *IEEE DSP Workshop*, September 1996.

[7] J. Proakis. *Digital Communications.* Mc Graw-Hill, 1989.

[8] F. Alberge, P. Duhamel, and M. Nikolova. Adaptive solution for blind Identification/Equalization using Deterministic Maximum Likelihood. *to be submitted IEEE Trans. on Signal Processing.*