

# ADAPTIVE BLIND EQUALIZATION USING WEIGHTED CUMULANT SLICES<sup>1</sup>

Josep Vidal and José A. R. Fonollosa

Department of Signal Theory and Communications, Universitat Politècnica de Catalunya  
08071 Barcelona, Spain  
Tel. (343) 4016457 Fax. (343) 4016447 E-mail: pepe@tsc.upc.es

## ABSTRACT

Many linear methods have been proposed in the literature to blindly estimate the ARMA parameters of a time series using HOS. Nevertheless, they are mainly off-line and not much has been done in the adaptive case. The method proposed in this contribution is the adaptive version of the w-slice method. The recursion is based on the inversion lemma when attempting the solution of an undetermined matrix equation. The system impulse response can be recovered regardless of the ARMA or MA character of the system. The number of operations depends on the square of the system order and it is considerably reduced with respect to previous approaches. Application to channel deconvolution is shown.

## 1. INTRODUCTION

Among the few adaptive methods, in [5] Swami and Mendel proposed a "Double Lattice" algorithm which was shown not to yield consistent estimates in a number of cases. On the other hand, Friedlander and Porat in [4] use the so-called "Overdetermined Recursive Instrumental Variable" (ORIV) [3] to compute recursive estimates of the Giannakis-Mendel (GM) method. The method solves an overdetermined system of equations using the inversion lemma. The drawbacks of the method arise from the use of 2nd and 3rd order statistics, which prevents from unbiased estimation in the case of colored Gaussian noise. Moreover, the method is valid only for the MA case. The method proposed below is the adaptive version of the w-slice method presented by these authors in [1]. The w-slice recursion is based on the inversion lemma when attempting the solution of an undetermined matrix equation.

## 2. UNDETERMINED RECURSIVE LEAST SQUARES

It was shown in [1] that the w-slice algorithm allows identification of the impulse response of any linear system in two steps:

<sup>1</sup> This work has been supported by the Spanish Government under grant TIC92-0800-C05-04.

$$S_a w = 1 \quad h_o = S_c w \quad (1)$$

where, if considering one 4th order bidimensional slice, matrices are defined as:

$$\begin{bmatrix} S_a \\ S_c \end{bmatrix} = \begin{bmatrix} C_{4x}(-p, -p, 0) & \dots & C_{4x}(-p, 0, 0) & \dots & C_{4x}(-p, p, 0) \\ \vdots & & \vdots & & \vdots \\ C_{4x}(-1, -p, 0) & \dots & C_{4x}(-1, 0, 0) & \dots & C_{4x}(-1, p, 0) \\ C_{4x}(0, -p, 0) & \dots & C_{4x}(0, 0, 0) & \dots & C_{4x}(0, p, 0) \\ C_{4x}(1, -p, 0) & \dots & C_{4x}(1, 0, 0) & \dots & C_{4x}(1, p, 0) \\ \vdots & & \vdots & & \vdots \\ C_{4x}(p, -p, 0) & \dots & C_{4x}(p, 0, 0) & \dots & C_{4x}(p, p, 0) \end{bmatrix}$$

A close view to this matrix reveals that the fourth-order equations (1) can be rewritten in the way:

$$\begin{aligned} X_t^T [Z_t - r_t(0)T_t - 2V_t] w &= 1 \\ h_o &= Y_t^T [Z_t - r_t(0)T_t - 2V_t] w \end{aligned} \quad (3)$$

where matrices X, Y and Z are defined as:

$$\begin{aligned} X_t^T &= \begin{bmatrix} 0 & \dots & 0 & x_0 \\ 0 & \dots & x_0 & x_1 \\ \vdots & & \vdots & \vdots \\ x_0 & \dots & x_{p-1} & x_p \\ \vdots & & \vdots & \vdots \\ x_{t-p} & \dots & x_{t-2p+1} & x_{t-2p} \end{bmatrix} & Y_t^T &= \begin{bmatrix} x_0 & x_1 & \dots & x_p \\ x_1 & x_2 & \dots & x_{p+1} \\ \vdots & \vdots & & \vdots \\ x_p & x_{p+1} & \dots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{t-p} & x_{t-p+1} & \dots & x_t \end{bmatrix} \\ Z_t &= \begin{bmatrix} 0 & \dots & 0 & x_0^3 & x_0^2 x_1 & \dots & x_0^2 x_p \\ 0 & \dots & x_1^2 x_0 & x_1^3 & x_1^2 x_2 & \dots & x_1^2 x_{p+1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ x_p^2 x_0 & \dots & x_p^2 x_{p-1} & x_p^3 & x_p^2 x_{p+1} & \dots & x_p^2 x_{2p} \\ x_{p+1}^2 x_1 & \dots & x_{p+1}^2 x_p & x_{p+1}^3 & x_{p+1}^2 x_{p+2} & \dots & x_{p+1}^2 x_{2p+1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ x_{t-p}^2 x_{t-2p} & \dots & x_{t-p}^2 x_{t-p-1} & x_{t-p}^3 & x_{t-p}^2 x_{t-p+1} & \dots & x_{t-p}^2 x_t \end{bmatrix} \end{aligned}$$

Matrices T and V account for the correlations terms in S:

$$\mathbf{T}_t = \begin{bmatrix} 0 & 0 & \dots & 0 & x_0 & \dots & x_{p-1} & x_p \\ 0 & 0 & \dots & x_0 & x_1 & \dots & x_p & x_{p+1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ x_{t-2p} & x_{t-2p+1} & \dots & x_{t-p} & x_{t-p+1} & \dots & x_{t-1} & x_t \end{bmatrix} \quad \mathbf{V}_t = \begin{bmatrix} x_0 r_p^T \\ x_1 r_{p+1}^T \\ \vdots \\ x_{t-p} r_t^T \end{bmatrix}$$

The autocorrelation vector  $\mathbf{r}$  must be computed recursively, so a possible choice is to use the following expression:

$$\mathbf{r}_t^T = \lambda \mathbf{r}_{t-1}^T + x_t \mathbf{u}_t^T$$

$$\mathbf{u}_t = [x_{t-p} \dots x_{t-1} x_t x_{t-1} \dots x_{t-p}]^T \quad (4)$$

All this matrices are updated as more data are available. The following notation is useful:

$$\mathbf{X}_{t+1} = [\mathbf{X}_t \quad \mathbf{x}_{t+1}] \quad \mathbf{Y}_{t+1} = [\mathbf{Y}_t \quad y_{t+1}]$$

$$\mathbf{Z}_{t+1} = \begin{bmatrix} \lambda \mathbf{Z}_t \\ \mathbf{z}_{t+1}^T \end{bmatrix} \quad \mathbf{T}_{t+1} = \begin{bmatrix} \lambda \mathbf{T}_t \\ \mathbf{t}_{t+1}^T \end{bmatrix} \quad \mathbf{V}_{t+1} = \begin{bmatrix} \lambda \mathbf{V}_t \\ \mathbf{v}_{t+1}^T \end{bmatrix} \quad (5)$$

$$\mathbf{N}_{t+1} = \mathbf{Z}_{t+1} - \mathbf{r}_{t+1} (0) \mathbf{T}_{t+1} - 2 \mathbf{V}_{t+1} = \begin{bmatrix} \lambda \mathbf{N}_t \\ \mathbf{n}_{t+1}^T \end{bmatrix} \quad (6)$$

The solution of equation (1) is done using pseudoinverse, leading to:

$$\mathbf{w}_m = \mathbf{S}_a^H \mathbf{1} = \mathbf{S}_a^H (\mathbf{S}_a \mathbf{S}_a^H)^{-1} \mathbf{1} = \mathbf{N}_t^H \mathbf{X}_t^H (\mathbf{X}_t \mathbf{N}_t \mathbf{N}_t^H \mathbf{X}_t^H)^{-1} \mathbf{1}$$

$$\mathbf{h}_o = \mathbf{Y}_t \mathbf{N}_t \mathbf{w}_m = \mathbf{Y}_t \mathbf{N}_t \mathbf{N}_t^H \mathbf{X}_t^H (\mathbf{X}_t \mathbf{N}_t \mathbf{N}_t^H \mathbf{X}_t^H)^{-1} \mathbf{1} \quad (7)$$

This algorithm has been shown to be asymptotically consistent in [1]. The derivation of the method is mainly based on how the inverse matrix in equation (7) is performed. Let there be:

$$\mathbf{P}_t = (\mathbf{X}_t \mathbf{N}_t \mathbf{N}_t^H \mathbf{X}_t^H)^{-1} \quad (8)$$

By substituting the relation (6) into (8) we will find a recursion for the inverse of  $\mathbf{P}$ :

$$\mathbf{P}_{t+1}^{-1} = \mathbf{X}_{t+1} \mathbf{N}_{t+1} \mathbf{N}_{t+1}^H \mathbf{X}_{t+1}^H =$$

$$= [\lambda \mathbf{X}_t \mathbf{N}_t + \mathbf{x}_{t+1} \mathbf{z}_{t+1}^T] [\lambda \mathbf{N}_t^H \mathbf{X}_t^H + \mathbf{z}_{t+1}^* \mathbf{x}_{t+1}^H] = \quad (9a)$$

$$= \lambda^2 \mathbf{P}_t^{-1} + \lambda \theta_{t+1} \mathbf{x}_{t+1}^H + \lambda \mathbf{x}_{t+1} \theta_{t+1}^H + \mathbf{x}_{t+1} \mathbf{z}_{t+1}^H \mathbf{z}_{t+1} \mathbf{x}_{t+1}^H =$$

$$= \lambda^2 \mathbf{P}_t^{-1} + \phi_{t+1} \Lambda_{t+1}^{-1} \phi_{t+1}^H$$

The use of the inversion lemma in equation (9):

$$\mathbf{P}_{t+1} = [\mathbf{P}_t - \mathbf{P}_t \phi_{t+1} (\lambda^2 \Lambda_{t+1} + \phi_{t+1}^H \mathbf{P}_t \phi_{t+1})^{-1} \phi_{t+1}^H \mathbf{P}_t] / \lambda^2 \quad (9b)$$

yields the complete recursion. Note that, unlike the approach in [4], an undetermined matrix equation is being solved, and hence, the resulting recursion differs greatly. Note also that equation (2) has been written by considering the use of  $2p+1$  slices, but the number of slices can be chosen provided that the minimum set which guarantees identifiability is included [6]. Moreover, the approach is valid to estimate the impulse response of any MA or ARMA linear system, even if the true orders are overestimated, as it has been proven in [6]. Both

features are very important since the exact nature of the underlying system is unknown in many applications. The total number of operations required depends the number of slices  $N$  used. When  $2p+1$  slices are taken into account, the total number of floating point operations is  $23p^2/2 + 83p/2 + 16$ . The minimum set of slices to guarantee identifiability have been found in [6]. If the underlying system is ARMA,  $p$  corresponds to the maximum of the AR order and MA order of the model. A recursive algorithm can be derived for the third order statistics case, following the same principles. In this case, matrix  $\mathbf{N}$  is modified so as to contain products of two samples of the input signal.

### 3. SQUARE-ROOT UNDETERMINED RECURSIVE LEAST SQUARES

The algorithm lacks of numerical stability as standard RLS. Note, however that the inversion lemma is computed as the sum of two non-positive definite matrices, and hence the symmetric characteristic of the inverse matrix  $\mathbf{P}$  will be lost if the numerical accuracy is not high enough. In this case divergence is sure and fast.

Square-root approaches depart from the update of the lower-triangular square-root of the matrix, instead of the matrix  $\mathbf{P}$  itself. This point is common to all square root methods. However, the proposed approach differs in that one of the auxiliary matrices is indefinite as was remarked in [2]. Because of this, orthogonal transformations (like Givens rotations) are not enough to update such matrices and hyperbolic transforms are necessary. Let us clarify this point first. It is a fact that:

$$\lambda^2 \Lambda_{t+1} = \begin{bmatrix} -\mathbf{z}_{t+1}^H \mathbf{z}_{t+1} & \lambda \\ \lambda & 0 \end{bmatrix} =$$

$$= \begin{bmatrix} \alpha & 0 \\ -\lambda/\alpha & \lambda/\alpha \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & -\lambda/\alpha \\ 0 & \lambda/\alpha \end{bmatrix} = \mathbf{A} \mathbf{J}_2 \mathbf{A}^T$$

where  $\alpha = (\mathbf{z}_{t+1}^H \mathbf{z}_{t+1})^{1/2}$ . Define the signature matrix of  $\Lambda_{t+1}$  as being  $\mathbf{J}$  and assume we can find a matrix  $\mathbf{Q}$  such that it fits the so-called  $J$ -orthogonal property:

$$\mathbf{Q} \mathbf{J}_{N+2} \mathbf{Q}^H = \mathbf{J}_{N+2} \quad (10)$$

with the additional feature of being the triangularizing operator of the matrix  $\mathbf{F}$ , that is:

$$\mathbf{F} \mathbf{Q} = \mathbf{L} \Leftrightarrow \begin{bmatrix} \mathbf{A} & \phi_{t+1}^H \mathbf{P}_t^{1/2} \\ 0 & \mathbf{P}_t^{1/2} \end{bmatrix} \mathbf{Q} = \begin{bmatrix} \mathbf{L}_1 & 0 \\ \mathbf{M} & \mathbf{L}_2 \end{bmatrix} \quad (11)$$

where  $\mathbf{L}_1$  and  $\mathbf{L}_2$  are lower triangular matrices of dimensions  $(2 \times 2)$  and  $(N \times N)$  respectively and  $\mathbf{J}_N$  is defined as:

$$\mathbf{J}_N = \begin{bmatrix} -1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{I}_{N-1} \end{bmatrix}$$

Then, by using (4.18), and right-multiplying both sides of equation (4.19) by  $\mathbf{J}_{N+2} \mathbf{L}^H$  we can rewrite:

$$F J_{N+2} F^H = L J_{N+2} L^H$$

Hence, by substituting terms in (4.19) and equating both sides, we have:

$$A J_2 A^H + \phi_{i+1}^H P_i \phi_{i+1} = \lambda^2 \Lambda_{i+1} + \phi_{i+1}^H P_i \phi_{i+1} = L_1 J_2 L_1^H \quad (12)$$

$$P_i \phi_{i+1} = M J_2 L_1^H \quad (13)$$

$$P_i = M J_2 M^H + L_2 L_2^H \quad (14)$$

Decomposition in equation (12) always exist since the signature of the matrix  $A J_2 A^H + \phi_{i+1}^H P_i \phi_{i+1}$  is  $J_2$  [6]. By taking (12) we get:

$$M = P_i \phi_{i+1} (L_1^H)^{-1} J_2$$

Substituting it in (14) and by similarity with the inversion lemma in equation (9b):

$$\begin{aligned} L_2 L_2^H &= P_i - M J_2 M^H = P_i - P_i \phi_{i+1} (L_1^H)^{-1} J_2 L_1^{-1} \phi_{i+1}^H P_i = \\ &= P_i - P_i \phi_{i+1} (\Lambda_{i+1} + \phi_{i+1}^H P_i \phi_{i+1})^{-1} \phi_{i+1}^H P_i = \lambda^2 P_{i+1} \end{aligned}$$

Hence, since  $L_2$  is lower triangular, it is proportional to the square-root of  $P_i$ , that is:

$$P_{i+1}^{1/2} = \lambda^{-1} L_2$$

Therefore, the process of updating the square-root matrix departs from the triangularization of  $F$ . Let us see how this may be achieved. Considering that matrix  $P_{i+1}^{1/2}$  is lower triangular,  $F$  has the following structure of zeros:

$$F = \begin{bmatrix} A & \phi_{i+1}^H P_i^{1/2} \\ 0 & P_i^{1/2} \end{bmatrix} = \begin{bmatrix} \times & 0 & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & \times & 0 & 0 & 0 \\ 0 & 0 & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix} \quad (15)$$

where the crosses denote possibly non-zero terms. The entries in the first row can be zeroed by using successive hyperbolic rotations [7]. This rotations have to be applied to cancel the non-zero elements of the first row in equation (15) because the  $J$ -orthogonality condition imposed to the triangularizing operator in equation (11). The rotations for the second row, must be orthogonal so that the complete triangularizing matrix be  $J$ -orthogonal. Orthogonal (also called Givens or Jacobi) rotations have the identity matrix as signature. The final URLS W-slice algorithm, written on its square root version, is shown in table 1, for the complex definition of the fourth order cumulant. The function  $J\_ort()$  performs the triangularization algorithm. Computational cost is not increased with respect to the non square-root version.

As a final comment, simulations on complex constellations signals have shown that performance is improved when using the complex definition of the fourth order cumulant:

$$C_{4,x}(0, m, n) = E\{x_i x_i^* x_{i+m}^* x_{i+n}\} - E\{x_i x_i^*\} E\{x_{i+m}^* x_{i+n}\} - E\{x_i x_{i+m}^*\} E\{x_i^* x_{i+n}\} - E\{x_i x_{i+n}^*\} E\{x_{i+m}^* x_{i+n}\}$$

#### 4.SIMULATIONS

Simulations are devoted to consider the recursive approach above in a blind equalization framework. A digital communications channel is equalized. In a 25 dB of SNR scenario, V-29 constellations are used. No synchronization error is supposed and the signal is sampled at one sample per symbol. The distorting channel is given by the following FIR impulse response:

$$H(z) = (0.944 - j0.87)[1 - (0.0787 + j0.0768)z][1 - (0.188 + j0.075)z^{-1}][1 + (0.158 + j0.426)z^{-1}]$$

Once the impulse response has been estimated, the MSE equalizer taps are computed. Figure 1 shows the tracking of the MSE, versus the number of adaptation steps, along with the tracing for the Benveniste-Goursat algorithm. All have been averaged over 15 independent realizations.

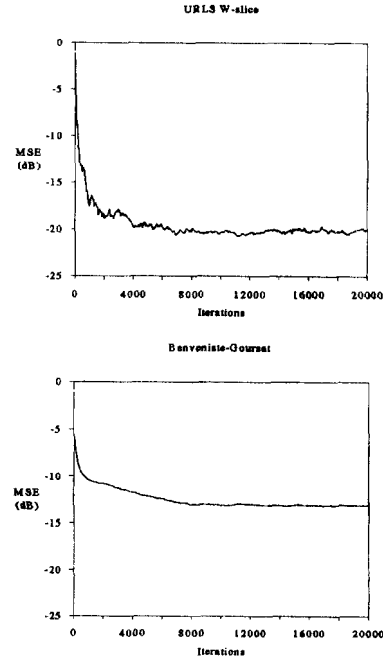


Figure 1. MSE tracings for the W-slice and BG equalizers.

Figure 2 displays the constellations prior and after equalization, at 10000 iterations. Note that as few as 4000 iterations are needed to open the eye in the w-slice case and decision-directed algorithms may then be used. On the other hand, the Benveniste-Goursat algorithm is far from convergence. It

should be stressed, however, that in convergence all decision-directed algorithms exhibit lower MSE. Phase rotations can also be compensated using the w-slice approach [6].

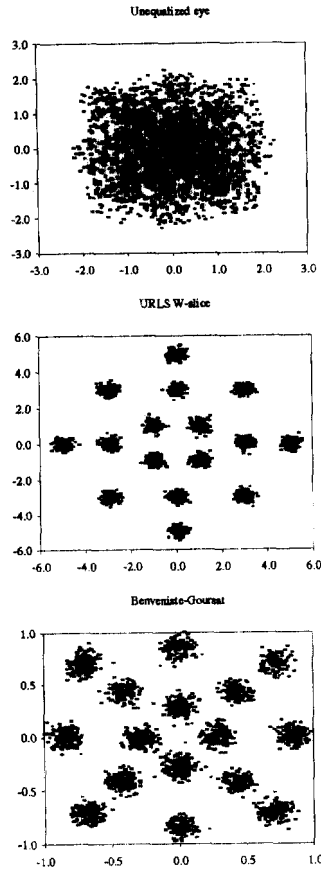


Figure 2. Eye patterns obtained for a V-29 constellation perturbed by a channel under 25 dB of SNR. Equalization by using URLS W-slice and BG algorithms after 10000 iterations.

#### REFERENCES

- [1] J.A.R. Fonollosa and J. Vidal, "System Identification Using a Linear Combination of Cumulant Slices", *IEEE Transactions on Signal Processing*, vol. 41, no. 7, July 1993.
- [2] B. Porat and B. Friedlander, "The Square-Root Overdetermined Recursive Instrumental Variable Algorithm", *IEEE Trans. on Automatic Control*, vol. AC-34, no. 6, June 1989.

- [3] B. Friedlander, "The Overdetermined Recursive Instrumental Variable Method", *IEEE Trans. on Automatic Control*, vol. AC-29, no. 4, April 1984.
- [4] B. Friedlander and B. Porat, "Adaptive IIR Algorithms Based on Higher-Order Statistics", *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, pp. 485-495, April 1989.
- [5] A. Swami and J. M. Mendel, "Adaptive System Identification Using Cumulants", *Proc. of the ICASSP'89*, Glasgow, pp. 2248-2251, May 1989.
- [6] J. Vidal, *Non-Gaussian ARMA Estimation Based on Higher-Order Statistics*, Ph. D. Dissertation, Universitat Politècnica de Catalunya, Barcelona, Spain, 1993.
- [7] G. H. Golub and C. F. VanLoan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1983.
- [8] J. Vidal and J.A.R. Fonollosa, "Adaptive Blind Identification Using Weighted Cumulant Slices", submitted to *Int. Journal of Adapt. Control & Signal Proc.*

Table 1. The complex fourth order URLS W-slice recursion

#### Initialization

$$S_0 = \mu [I_{p+1} \mid 0]$$

$$P_0 = \frac{1}{\mu^2} I_{p+1}$$

$$L_0 = \mu [I_{p+1} \mid 0]$$

For  $t = 0, 1, 2, \dots$

$$u_{t+1} = [x_{t-(N-1)/2+1} \dots x_{t+1} \dots x_{t-(N-1)/2+1}]^T$$

$$y_{t+1} = [x_{t-p+1} \dots x_{t+1}]^T$$

$$x_{t+1} = [x_{t-2p+1} \dots x_{t-p+1}]^T$$

$$r1_{t+1} = \lambda r1_t + x_{t+1} u_{t+1}^*$$

$$r2_{t+1} = \lambda r2_t + x_{t+1} u_{t+1}$$

$$n = \min(t, 1/(1-\lambda))$$

$$z_{t+1} = [x_{t-(N-1)/2+1} \dots x_{t+1}]^T$$

$$t_{t+1} = [x_{t-N+2} \dots x_t \ x_{t+1}]^T$$

$$n_{t+1} = z_{t+1} - (r_{t+1}(0)t_{t+1}^* + x_{t-p+1}r2_{t+1}^* + x_{t-p+1}^*r1_{t+1})/n$$

$$\theta_{t+1} = S_t n_{t+1}^*$$

$$L_{t+1} = \lambda L_t + y_{t+1} n_{t+1}^T$$

$$S_{t+1} = \lambda S_t + x_{t+1} n_{t+1}^T$$

$$\phi_{t+1} = [\theta_{t+1} \ x_{t+1}]^T$$

$$\alpha = \sqrt{n_{t+1}^H n_{t+1}}$$

$$A = \begin{bmatrix} \alpha & 0 \\ -\lambda/\alpha & \lambda/\alpha \end{bmatrix}$$

$$R = \begin{bmatrix} A & \phi_{t+1}^H P_t^{1/2} \\ 0 & P_t^{1/2} \end{bmatrix}$$

$$Q = J_{-} \text{ort}(R) = \begin{bmatrix} L_t & 0 \\ M & \lambda P_{t+1}^{1/2} \end{bmatrix}$$

$$h'_{t+1} = L_{t+1} S_{t+1}^H P_{t+1}^{1/2} (P_{t+1}^{1/2})^H 1$$