Hammerstein and Volterra Adaptive Blind Equalization for Wiener Channels

Carlos A. R. Fernandes, Gérard Favier and João Cesar M. Mota

Abstract— **Currently, the most part of the adaptive techniques to perform equalization of nonlinear channels on the literature is trained. This works proposes, in an original way, adaptive techniques to perform blind equalization of nonlinear channels, specifically with Wiener structures. To do so, we are going to make use of the Constant Modulus Algorithm (CMA). We develop expressions for the adaptation of the equalizer using a unified notation for three different equalizer filter structures: i) an Hammerstein filter; ii) a diagonal Volterra filter; and iii) a Volterra filter. Trying to improve the performance of the CMA equalizers, we will also employ the step-size normalization of these algorithms and develop Recursive Least Squares (RLS) version of them.**

Keywords— **Adaptive Blind Equalization, Wiener Model, Hammerstein Model, Volterra Model, Constant Modulus Algorithm**

Resumo— **As técnicas atualmente encontradas na literatura para realização de equalização adaptativa de canais não-lineares são, em sua grande maioria, supervisionadas. O presente trabalho propõe, de forma original, algoritmos cegos e adaptativos para equalização de canais não-lineares, especificamente canais do tipo Wiener. Para tanto, vamos fazer uso do Algoritmo do Módulo Constante (CMA). As expressões para a adaptação do equalizador são encontradas com o auxílio de uma notação unificada para três diferentes estruturas: i) um filtro de Hammerstein; ii) um filtro de Volterra diagonal; e iii) um filtro de Volterra completo. Com o intuito de melhorar a performance dos equalizadores não-lineares do tipo CMA, este trabalho também propõe versões do CMA normalizadas e baseadas no algoritmo dos Mínimos Quadrados Recursivos (RLS) para o caso de canais não-lineares.**

Palavras-Chave— **Equalização Autodidata Adaptativa, Modelo de Wiener, Modelo de Hammerstein, Modelo de Volterra, Algoritmo do Modulo Constante.**

I. INTRODUCTION

Many nonlinear systems can be modelled as a cascade of linear blocks with memory and memoryless nonlinearities. Particularly, the Wiener model, which consists of a linear block with memory followed by a memoryless nonlinearity, is very used in communications systems.

The Wiener filters can model, for example, satellite communication channels, in which high power amplifiers are driven at or near saturation, to achieve the power consumptions requirements. The result is the introduction of nonlinear band-limited signal distortion. Another important application of Wiener models is in the modelling of Radio Over Fiber (ROF) links in communications systems. In this case, the signal is transmitted by a mobile station and it is converted in optical frequencies in a Radio Access Point (RAP), and then transmitted through optical fibers. When the length of the optical fiber is short (order of kilometers) and the radio frequency have an order of GHz, the dispersion of the of the fiber is negligible. In this case, the nonlinear distortion arising from the electrical to optical conversion process becomes preponderant. Thus, the system can be seen as a Wiener model. Application of these nonlinear models are also present in other domains, like control valves and biological systems.

There has been several works that propose and study techniques for identification and equalization of nonlinear communication channels. Some of the first works on nonlinear equalization of communication channels were done in [1], [2], [3], most of them using Volterra filters. Most recently, Hedge et al. in [4] studied the identification of a series cascade of a linear FIR filter, followed by a memoryless nonlinearity and followed by a second linear FIR filter (Wiener-Hammerstein system). The filters are adapted by the Normalized Least Mean Square (NLMS) algorithm in two different filter identification structures: a Wiener-Hammerstein structure and a Volterra filter followed by an Moving Average (MA) linear filter.

Another interesting work in this area was done in [5], where the authors have also used an equalizer with a Volterra structure to compensate the distortions of nonlinear Satellite Channels. In this work, they used a diagonal matrix with different step-size values among the diagonal instead of a fixed step-size. This approach was found to improve the performance of the LMS-Volterra equalizer. Another way to improve the performance of nonlinear equalizers was found by Fernando et al. in [6]. They proposed an equalizer filter structure composed of a Hammerstein filter followed by a Decision Feedback Equalizer (DFE). The simulation results showed that the DFE inclusion in nonlinear filters seems to be a good option for the improvement of nonlinear equalizers performance.

However, the adaptive techniques to perform nonlinear blind equalization in the literature are, in the most part, trained. The main contribution of this work is to propose blind adaptive techniques to equalize channels with Wiener structures. To do so, we are going to make use of the CMA, one of the most used algorithms for blind equalization of linear channels. We are going to develop the expressions for the adaptation of the equalizer using three different approaches, each one considering a different structure for the equalizer. The first one considers the nonlinear communication system model of in fig. 1, which shows a Hammerstein-type equalizer for a Wiener-type channel. It is known that the inverse of a Wiener system

Carlos A. R. Fernandes and Gérard Favier, *Laboratoire I3S, Université de Nice - Sophia Antipolis*, Nice, France. E-mails: acarlos@i3s.unice.fr, favier@i3s.unice.fr.

João Cesar M. Mota, Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará, Fortaleza, Brasil. E-mail: mota@deti.ufc.br.
This work was partially financed by ALBAN.

is a Hammerstein system [7], so it is the first natural choice for the equalization of a Wiener channel. This approach considers the independency of the linear and the nonlinear filters to adapt them separately. The second approach adapts these two filters jointly, constituting a Diagonal Volterra filter. And finally, the last approach considers the equalizer as a complete Volterra filter. Trying to improve the performance of the nonlinear CMA equalizers, we will employ the step-size normalization of these algorithms and develop RLS version of them in the next chapter. All these techniques are develop using an unified notation introduced in the next section. Moreover, it was possible to write the three nonlinear equalizer structures in a way that they are linear with respect to their coefficients.

The rest of the work is organized as follows: section II briefly explains the structure of the system and develop a unified signal notation; section III develops the adaptation expressions of the considered equalizers based on the CMA, the NCMA (Normalized CMA) and the RCMA (Recursive CMA); section IV illustrates the performance of the algorithms by means of computational simulations; and some conclusion and perspectives are drawn in section V.

II. SIGNAL MODELS

A simplified version of the nonlinear SISO (Single-Input Single-Output) communication system model employed in this work is shown in fig. 1. The channel is modelled by a linear Moving-Average (MA) filter followed by a polynomial filter, i.e., a Wiener model. The equalizer structure is composed by a polynomial filter followed by a linear MA filter, i.e., an Hammerstein model. We are also going to use a complete and a diagonal Volterra filter structure for the equalizer. The transmitted i.i.d. sequence $\{a(n)\}$ can take the value of any constellation symbol with equal probability. The output of the linear part of the channel $u(n)$ can be expressed by:

$$u(n) = \sum_{i=0}^{N-1} a(n-i)h_i = \mathbf{h}^T\mathbf{s}(n),$$

where $\mathbf{h} = [h_0 \; h_1 \; . \; . \; . \; h_{N-1}]^T$ is the impulse response of the linear part of the channel, N is the length (memory) of $\mathbf{h}$ and $\mathbf{a}(n) = [a(n) \; a(n-1) \; . \; . \; . \; a(n-N+1)]^T$ is the vector containing the transmitted sequence. It is important to note that $u(n)$ is not an accessible signal. The received signal $x(n)$ can be expressed as in (1) where $\mathbf{c} = [c_0 \; c_1 \; . \; . \; . \; c_{L-1}]^T$ is the vector containing the weights of the nonlinearity of the channel, L is the length (order) of the nonlinear part of the channel, $\mathbf{u}(n) = [1 \; u(n) \; u^2(n) \; . \; . \; . \; u^{L-1}(n)]^T$ is the vector containing the inputs of the nonlinear part of the channel and $v(n)$ is an additive white Gaussian noise (AWGN) component.

$$x(n) = \sum_{i=0}^{L-1} c_i u^i(n) + v(n) = \mathbf{c}^T\mathbf{u}(n) + v(n). \quad (1)$$

The output of the nonlinear part of the equalizer $z(n)$ and the final output of the equalizer $y(n)$ are given by eqs. (2) and (3), respectively:

$$z(n) = \sum_{i=0}^{P-1} g_i x^i(n) = \mathbf{g}^T\mathbf{x}(n), \quad (2)$$

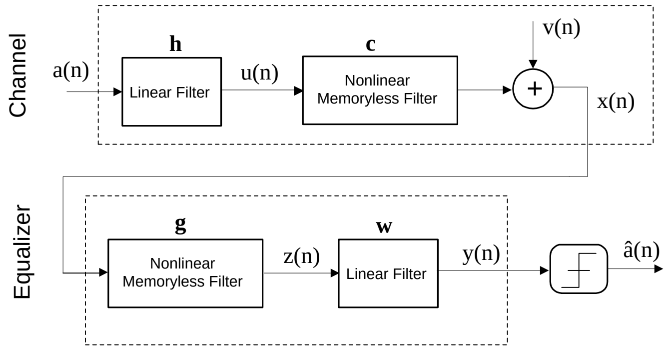

Fig. 1. Simplified structure of the system.

$$y(n) = \sum_{i=0}^{M-1} z(n-i)w_i = \mathbf{w}^T\mathbf{z}(n), \quad (3)$$

where P is the length (order) of the nonlinear part of the equalizer, $\mathbf{g} = [g_0 \; g_1 \; . \; . \; . \; g_{P-1}]^T$ is the tap-weight vector of the nonlinear part of the equalizer, $\mathbf{x}(n) = [1 \; x(n) \; x^2(n) \; . \; . \; . \; x^{P-1}(n)]^T$ is the vector containing the inputs of the nonlinear part of the equalizer, M is the length (memory) of the linear part of the equalizer, $\mathbf{w} = [w_0 \; w_1 \; . \; . \; . \; w_{M-1}]^T$ is the tap-weight vector of the linear part of the equalizer and $\mathbf{z}(n) = [z(n) \; z(n-1) \; . \; . \; . \; z(n-M+1)]^T$ is the vector containing the inputs of the linear part of the equalizer.

We can also write the equalizer output in a compact way, by substituting (2) in (3):

$$y(n) = \sum_{i=0}^{M-1} \sum_{j=0}^{P-1} g_j w_i x^j(n-i) \Rightarrow \quad (4)$$

$$y(n) = \mathbf{g}^T\mathbf{X}(n)\mathbf{w},$$

where $\mathbf{X}(n) = [\mathbf{x}(n) \; \mathbf{x}(n-1) \; . \; . \; . \; \mathbf{x}(n-P+1)]$. If we call $\mathbf{p}(n) = \mathbf{X}(n)\mathbf{w}$, we can also express the equalizer output by:

$$y(n) = \mathbf{g}^T\mathbf{p}(n). \quad (5)$$

Equations (3) and (5) will be used in the development of the algorithms by optimizing the two equalizer filters $\mathbf{g}$ and $\mathbf{w}$ in a alternating way. This approach will be developed in the next section and it supposes the independency between the filters to find two different expressions for the adaptation of $\mathbf{g}$ and $\mathbf{w}$. The algorithms will be also developed by an approach that adapts the equalizer filters jointly. This approach consider the equalizer as having a Volterra filter structure. We will test a complete and a diagonal Volterra filter to compare their different performances. From (4), we can express the equalizer output by:

$$y(n) = \mathbf{r}^T\tilde{\mathbf{x}}(n), \quad (6)$$

where $\tilde{\mathbf{x}}(n) = [1 \;\; x(n) \;\; x^2(n) \; . \; . \; . \; x^{P-1}(n) \;\; 1 \;\; x(n-1) \;\; x^2(n-1) \; . \; . \; . \; x^{P-1}(n-1) \; \cdots \; 1 \;\; x(n-M+1) \;\; x^2(n-M+1) \; . \; . \; . \; x^{P-1}(n-M+1)]^T$ and $\mathbf{r} = [r_0 \; r_1 \; . \; . \; . \; r_{PM-1}]^T = \mathbf{g} \otimes \mathbf{w}$ ($\otimes$ represents the Kronecker product). The vector $\mathbf{r}$ can also be expressed by $r_{j+iP} = g_j w_i$, where $0 \leq j \leq P-1$ and $0 \leq i \leq M-1$. The vector **r** contains the coefficients of the linear and the nonlinear parts of the equalizer and it is a particular case of the Volterra filter (diagonal Volterra).

If we model the equalizer as a complete Volterra filter, its output can be written as expressed next:

$$y(n) = \sum_{i=0}^{P-1} \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} \cdots \sum_{n_p=0}^{M-1} b_{(i,n_1,\cdots,n_p)} \prod_{j=1}^{i} x(n-n_j), \quad (7)$$

where $b_{(i,n_1,\cdots,n_p)}$ are the Volterra coefficients, M is the memory of the model and P is the order. We can express (7) in a vectorial form:

$$y(n) = \mathbf{b}^T \check{\mathbf{x}}(n), \quad (8)$$

where **b** is a vector containing the Volterra coefficients and $\check{\mathbf{x}}(n) = [\check{\mathbf{x}}_1^T(n) \;\; \check{\mathbf{x}}_2^T(n) \;\; \cdots \;\; \check{\mathbf{x}}_P^T(n)]^T$ is the vector containing all the inputs of the Volterra filter. The input vector $\check{\mathbf{x}}(n)$ is obtained from the vector $\check{\mathbf{x}}_1(n) = [x(n) \; x(n-1) \; . \; . \; . \; x(n-M+1)]^T$, by using the relation $\check{\mathbf{x}}_P(n) = \check{\mathbf{x}}_1(n) \otimes \cdots \otimes \check{\mathbf{x}}_1(n)$ ($P-1$ times the Kronecker product).

Unified Notation: A New Approach

We can simplify the above notation by defining an unified notation that describes the equations (3), (5), (6) and (8). To do so, we must write for the equalizer output:

$$y(n) = \mathbf{q}^T \mathbf{t}(n), \quad (9)$$

where **q** is the equalizer coefficients and $\mathbf{t}(n)$ is the vector containing the equalizer inputs. We can see (9) as general equation for $y(n)$ where the vectors **q** and $\mathbf{t}(n)$ depend on the technique used. Table I shows these values for the techniques studied here. This unified notations described in (9) will guide us to develop different techniques using the same mathematical model.

TABLE I

UNIFIED NOTATION

Approach	$\mathbf{q}$	$\mathbf{t}(n)$
Separated - **g** Adaptation	$\mathbf{g}$	$\mathbf{p}(n)$
Separated - **w** Adaptation	$\mathbf{w}$	$\mathbf{z}(n)$
diagonal Volterra	$\mathbf{r}$	$\tilde{\mathbf{x}}(n)$
Volterra	$\mathbf{b}$	$\check{\mathbf{x}}(n)$

It should be highlighted that, using this unified notation, the output of the equalizer $y(n)$ is linear with respect to coefficients of the the filters. That means that the techniques that use linear structures can be applied directly in the nonlinear filter structures considered in this work.

III. Adaptive Blind Nonlinear Equalization

The unified notation described by eq. (9) is used in this section to develop the algorithm expressions for the different approaches. The first one considers the independency of the filters **g** and **w** to adapt them separately and in a alternating way by using the eqs. (3) and (5). The other two uses eqs. (6) and (8) to adapt the filters **r** or **b**.

Before studying nonlinear blind equalization by using the CMA, we will first apply the Least Mean Square (LMS) algorithm in the considered equalizer structures. This will guide us to develop the nonlinear CMA and to compare the trained and the blind algorithms.

A. LMS Adaptation

To develop the LMS expression using the different approaches, we may take the gradient of the Minimum Mean Square Error (MMSE) cost function with relation to **q** (eq. (9)). The MMSE cost function is given by:

$$J_{MMSE} = \mathrm{E}\{|e(n)|^2\} = \mathrm{E}\{|d(n) - y(n)|^2\},$$

where $d(n)$ is the desired signal. By taking the stochastic gradient of J_{MSE} with respect to **q**, we may find the LMS adaptation expression for the three approaches:

$$\mathbf{q}(n+1) = \mathbf{q}(n) + \mu e(n) \mathbf{t}^*(n), \quad (10)$$

where **q** and **t** are given in Table I. The step-size parameter μ is not necessarily the same in all approaches.

The LMS-Separated technique adapts the **g** and **w** filters separately and in an alternating way. The equalizer adaptation using the LMS-Separated technique is the less complex, with an order of complexity of $O(M+P)$, and it is done alternating the adaptations of **g** and **w**. t is the less complex of the approaches, with an order of complexity of $O(M+P)$. The order of complexity of the LMS-Diagonal Volterra and LMS-Volterra are $O(M.P)$ and $O(\frac{M^{P+1}-M}{M-1})$, respectively.

B. Nonlinear CMA

We may develop the techniques to equalize adaptively and blindly nonlinear channels by applying the CMA to the considered structures. For the development of the CMA with a Hammerstein, a diagonal Volterra and a Volterra structure, we will proceed the same way as explained earlier. That means we take the stochastic gradient of the CM cost function, given by:

$$J_{CM} = \mathrm{E}\{(R - |y(n)|^2)^2\}, \quad (11)$$

with relation to **q**, where R is a constant given by $\frac{\mathrm{E}\{|a(n)|^4\}}{\mathrm{E}\{|a(n)|^2\}}$. Thus, by using the Stochastic Gradient Descent approach, we find an unified CMA adaptation equation for all the approaches:

$$\mathbf{q}(n+1) = \mathbf{q}(n) + \mu y(n)(R - |y(n)|^2) \mathbf{t}^*(n). \quad (12)$$

The values of **q** and **t** in Table I defines the different techniques. Again, the step-size parameter μ is not necessarily the same in equations for the different techniques. In the CMA-Separated technique, the adaptations of **g** and **w** are also done in an alternating way. The order of complexity of these CMA techniques is the same of LMS techniques. It is important to remark that adaptive blind algorithms to perform equalization of nonlinear channels are not very common in the literature.

For both CMA and LMS algorithms, despite of the Volterra approaches have only one adaptation equation, their complexities are much bigger than for the separated approach. Moreover, the high number of parameters to be adapted in the Volterra approaches makes their convergence speeds slow, once the bound of the step size is inversely proportional to the equalizer length.

C. Step-size Normalization

For the equalization of linear channels, the employment of the step-size normalization is often a good choice to increase the convergence rate of the LMS-type algorithms. Due to this, in this section we will develop normalized versions of the techniques proposed in the last section. To do so, we will make use of the NCMA, which is based on a particular choice of the step-size. At each iteration we choose a step-size such that the updated filter coefficients achieve the desired modulus when applied to current data vector. In our case, the goal is to define an unified optimization problem for the three different approaches: the Separated, the Diagonal Volterra and the Volterra. In the sequel, we will write the cost function and the constraints associated with this cost function in an unified way. Thus, the development of the three techniques is done identically.

By using the unified notation, we can express the Normalized CM (NCM) cost function as showed next:

$$J_{NCM} = \|\delta \mathbf{q}(n)\|^2 = \|\mathbf{q}(n+1) - \mathbf{q}(n)\|^2.$$

The minimization of this cost functions must respect the following energy constraint:

$$|\mathbf{q}^T(n+1)\mathbf{t}(n)|^2 = R,$$

where R is the same constant of the CMA. The solution to this optimization problem is developed in an identical way of the development of the NCMA in a linear structure [8], leading us to:

$$\mathbf{q}(n+1) = \mathbf{q}(n) - \frac{\mu}{\|\mathbf{t}(n)\|^2} y(n)(\frac{\sqrt{R}}{|y(n)|} - 1)\mathbf{t}^*(n),$$

where the values of **q** and **t** in Table I defines the different techniques.

As well as for the LMS and CMA, the step-size parameter μ is not necessarily the same for the different techniques and the adaptations of **g** and **w** in the Separated approach are done in an alternating way. The order of complexity of these NCMA techniques is the same than for the LMS.

As we said, the NCMA is based on a particular choice of the step-size. In case of the separated adaptation, to adapt the nonlinear filter, at each iteration we choose a step-size such that the updated **g** filter achieves the desired modulus when applied to current data vector and to the non-updated **w** filter. To adapt the linear filter, we have the opposite situation, the step-size is chosen such that the updated **w** filter achieves the desired modulus when applied to current data vector and to the non-updated **g** filter. In the Volterra cases these choices are trivial once there is only one filter to adapt.

D. RCMA Adaptation

Trying to produce a very fast-converging adaptive blind algorithms at the expense of increased complexity, we may develop RLS versions of the algorithms presented earlier for equalization of Wiener channels. To do so, we will make use of the NCMA, which, in this case, must recursively minimize the following cost function:

$$\phi(n) = \sum_{i=0}^{n} \lambda^{n-i}(|y(i)|^2 - R), \tag{13}$$

where $\lambda \leq 1$ is the forgetting factor and the value $y(n)$ is given by $y(i) = \mathbf{q}^T(n)\mathbf{t}(i)$. That means that the filter we want to minimize **q** is always indexed by n in equation (13) and the filter inputs **t** by i. The definition of the different values of $y(n)$ in equation (13) was inspired in the definition of the RLS cost function. These values define different cost functions and, consequently, different algorithms.

The solution to this optimization problem is developed in an identical way of the development of the Recursive CMA (RCMA) in [8]. It can be summarized by the following equations:

$$\mathbf{s}(n) = \xi^*(n)\mathbf{t}(n),$$

$$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{s}^*(n)}{\lambda + \mathbf{s}^\top(n)\mathbf{P}(n-1)\mathbf{s}^*(n)},$$

$$\mathbf{P}(n) = \lambda^{-1} \cdot \left[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{s}^\top(n)\mathbf{P}(n-1)\right],$$

$$\mathbf{q}(n) = \mathbf{q}(n-1) + \mathbf{k}(n) \cdot \left(|\xi(n)|^2 - R\right),$$

where $\mathbf{P}(n)$ is initialized as $\mathbf{P}(0) = \delta^{-1}\mathbf{I}_{N_q}$, δ is a small positive constant, $\mathbf{I}_{N_q}$ is the N_q-by-N_q identity matrix, N_q is the length of **q** and $\xi(n) = \mathbf{q}^T(n-1)\mathbf{t}(n)$ is the *a priori output estimation*. Each cost function (eq. 13) is characterized by a pair **q-t** which depends on the filter we want to minimize, leading, for each case, to a different algorithm.

As well as for the case of linear channels, the RCMA can improve the convergence speed and the steady-steady error of the CMA and NCMA-type algorithms, which, in some cases, may not have acceptable values.

IV. Simulation Results

The proposed techniques were tested by means of computational simulations and they successfully performed adaptive blind equalization of nonlinear channels. The simulation scenario parameters used in the simulations are shown in Table II. The linear channel impulse response **h** and the nonlinear channel polynomial coefficients **c** are the same of [9]. All the Mean Squared Error (MSE) curves were obtained via Monte Carlo simulations using 100 independent data realizations.

TABLE II

Simulation Parameters

Linear Channel	$\mathbf{h} = [1 \;\; 0.5 \;\; -0.2]^T$
Nonlinear Channel	$\mathbf{c} = [0 \;\; 1 \;\; 0.3 \;\; 0.1]^T$
SNR	30 dB
M	4
P	6
Modulation	BPSK

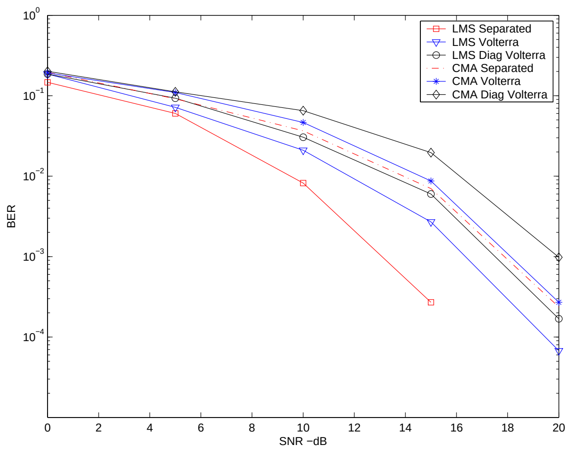

Fig. 2. BER curves for the CMA and LMS algorithms.

Fig. 2 shows the Bit Error Rate (BER) for the CMA and LMS adaptations (LMS in training mode). The separated approach has shown to be the better for both cases and the Diagonal Volterra approach the worst. To better analyze the convergence of the algorithms, fig. 3 shows MSE evolution of CMA and LMS algorithms using the separated and the Volterra approaches. The first remark we must do is the convergence of both CMA approaches. Moreover, we can see in Table III, which shows the number of parameters of the equalizer for all approaches, that the Volterra approaches have a bigger number of parameters than Separated approach. The big number of parameters makes the Volterra convergence speed slow, as we can see in the fig. 3. The large number of information of the Volterra approaches is prejudicial to the performance of the CMA. With relation to the CMA, the LMS has an error level of approximately 5dB smaller for the Separated approach and 3dB for the Volterra approach.

TABLE III

Number of Parameters of the Equalizer

Technique	N_q
Separated	10
Volterra diagonal	24
Volterra	258

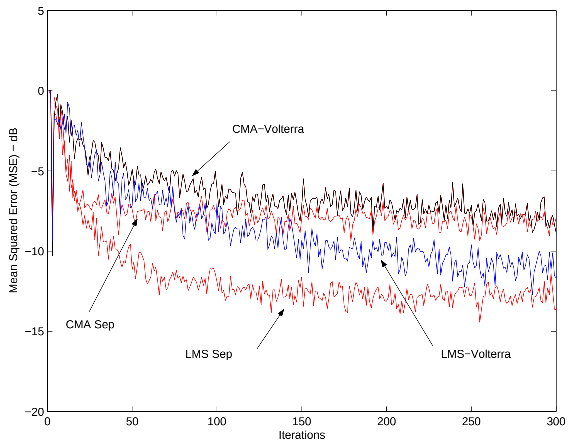

Fig. 3. MSE curves for the CMA and LMS algorithms.

The next results are related to the Normalized algorithms. Fig. 4 shows the BER for the NCMA and NLMS algorithms (NLMS in training mode). The separated approach has shown to be the better for both algorithms, the Diagonal Volterra the worst for the NCMA and the Volterra the worst for the NLMS. In this case, the BER performances are not as different as in the case of the LMS-type algorithms. In addition, we can see in the MSE curves of the algorithms (fig. 5) their very similar performances and the 4dB gain provided by the NLMS-Separated technique in relation to the others.

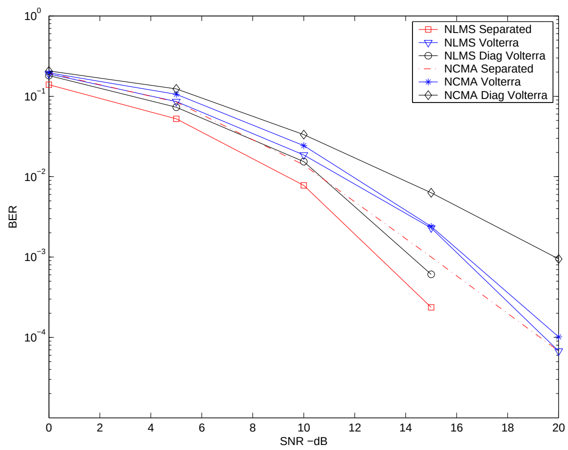

Fig. 4. BER curves for the NCMA and NLMS algorithms.

Similar results were obtained for the RLS-type algorithms. In fig. 6, we can be see the BER for the RCMA and RLS algorithms (RLS in training mode). Again, the separated approach

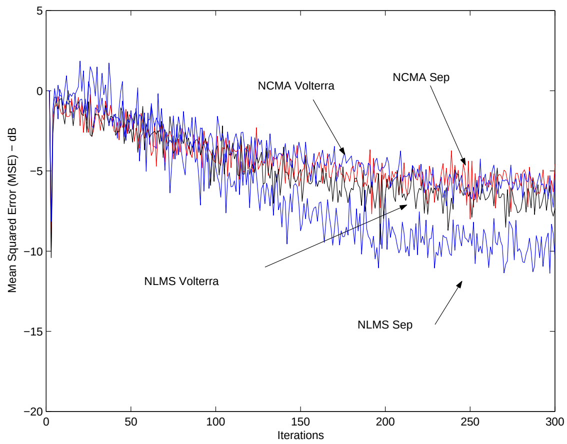

Fig. 5. MSE curves for the NCMA and NLMS algorithms.

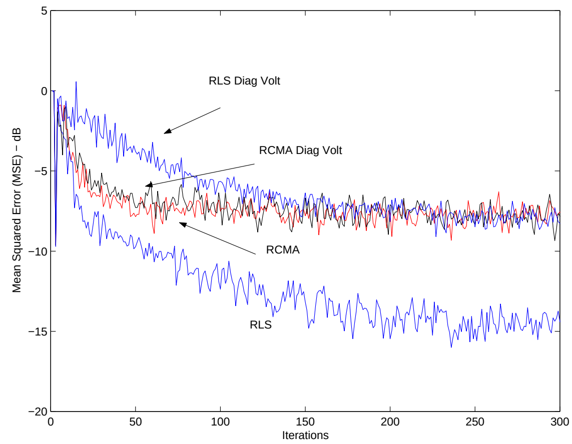

Fig. 7. MSE curves for the RCMA and RLS algorithms.

has shown to be the best for both algorithms. Finally, in fig. (7) we must also remark the convergence of all approaches and the better performance of the Separated approach with respect to the Diagonal Volterra for the RLS algorithm. The RCMA and the RLS have an error level approximately equal in the Diagonal Volterra approach, and the RLS has a 5dB gain in MSE in the separated approach.

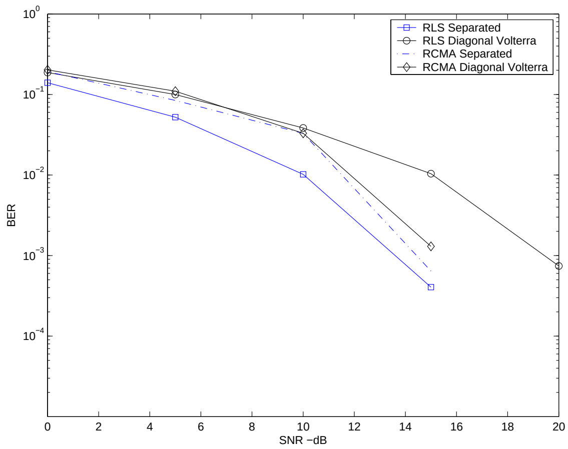

Fig. 6. BER curves for the RCMA and RLS algorithms.

V. CONCLUSIONS

This works brings some important contributions. The first one is the proposition of adaptive blind algorithms for equalization of Wiener channels that have shown to correctly recover the symbols passed throw the tested channel. Another important contribution was the comparison of different structures to adapt the equalizer. This was possible due to a unified notation introduced, in which the output of the equalizer is linear with respect to coefficients of the the filters. That means that the techniques that use linear structures were applied directly in the nonlinear filter structures considered in this work. The best simulations results were obtained when we adapt the linear and the nonlinear filters separately and in an alternating way. We also developed NCMA and RCMA techniques for the equalization of Wiener channels. They also have shown to correctly recover the symbols passed through the tested channel and, once again, the best simulations results were obtained by the Separated approach.

As perspectives of this work, we must tested the algorithms in other kinds of channels and use Volterra models jointly with other approaches developed in our research group [9]. As an example we can cite the application of the Parallel Factor (PARAFAC) analysis decomposition in systems modelled by Volterra series and other kind of diversity, as multiple sensors, redundancy induced by a linear precoder, oversampling etc.

REFERENCES

[1] D. D. Falconer, *Adaptive equalization of channel nonlinearities in QAM data transmission*. Bell Syst. Tech. J., v. 57, n. 7, p. 2589-2611, Sep. 1978.

[2] S. Benedetto and E. Biglieri, *Nonlinear equalization of digital satellite channels*. IEEE J. Select. Areas Commun., v. 1, n.1, p. 57-62, 1983.

[3] G. Karam and H. Sari, *Analysis of predistortion, equalization, and IS1 cancellation techniques in digital radio systems with nonlinear transmit amplifiers*. IEEE Trans. Commun., v. 37, p. 1245-1253, Dec. 1989.

[4] V. Hedge, C. Radhakrishnan, D. Krusienski and W. K. Jenkins, *Series-Cascade Nonlinear Adaptive Filters*. 45th Midwest Symposium on Circuits and Systems., v. 3, p. 219 – 222, 4-7 Aug. 2002.

[5] A. Gutierrez and W. E. Ryan, *Performance of Adaptive Volterra Equalizers on Nonlinear Satellite Channels*. IEEE International Conference on Communications, Seattle, v. 1, p. 488-492, 18-22 Jun 1995.

[6] Xavier N. Fernando and Abu. B. Sesay, *A Hammerstein type equalizer for the Wiener type fiber-wireless channel*. IEEE Pacific Rim Conf. on Comm., Computers and signal Processing, v.2, p. 546–549, 26-28 Aug. 2001.

[7] S. A. Billings and S. Y. Fakhouri, *Identification of systems containing linear dynamic and static nonlinear elements*. v. 18, n. 1, p. 15–26, Automatica, 1982.

[8] R. Pickholtz and K. Elbarbary, *The recursive constant modulus algorithm: a new approach for real time array processing*. ASILOMAR Conf. on Signals, Systems and Computers, n. 27, p. 627–632, 1993.

[9] A. Y. Kibangou, *Modeles de Volterra a Complexité Reduite: Estimation Paramétrique et Application a l'Egalisation des Canaux de Communication* . Universite de Nice Sophia Antipolis - UFR Sciences - France(*in french*), PhD Thesis, 2005.