

# NEW ADAPTIVE BLIND EQUALIZATION ALGORITHMS FOR CONSTANT MODULUS CONSTELLATIONS

Constantinos B. Papadias

Dirk T. M. Slock

Institut EURECOM, 2229 route des Crêtes  
B.P. 193, 06904, Sophia Antipolis Cedex, FRANCE

## ABSTRACT

We present a new class of adaptive filtering algorithms for blind equalization of constant modulus signals. The algorithms are first derived in a classical system identification context by minimizing at each iteration a deterministic criterion and then their counterpart for blind equalization is derived by modifying this criterion taking into account the constant-modulus property of the transmitted signal. The algorithms impose more constraints than the classical Constant Modulus Algorithm (CMA) and as a result achieve faster convergence. An asymptotic analysis has provided useful parameter bounds that guarantee the algorithms' stability. *A priori* knowledge of these bounds helps the algorithms escape from undesirable local minima of their cost function thus giving them a potential advantage over the classical CMA. An efficient computational organization for the derived algorithms is also proposed and their behaviour has been tested by means of computer simulations.

## 1. INTRODUCTION

We consider the classical blind equalization problem described in figure 1 where a baseband representation of a communication system is shown. The symbols  $\{a_k\}$  are considered to be an i.i.d. sequence belonging to a discrete alphabet of constant modulus and are transmitted through a linear channel. The received sequence  $\{x_k\}$  (corrupted also by the additive noise  $\{\eta_k\}$ ) is passed through a linear (FIR) equalizer that is updated by a blind equalization algorithm in order to open the communication system's eye and result in a correct retrieval of the transmitted sequence with the help of the decision device. The equalizer's output at time instant  $k$  is denoted by  $y_k$  and may be written as  $X_k^H W_k$ , where  $X_k = [x_k \ x_{k-1} \ \dots \ x_{k-N+1}]^H$ ,  $W_k$  is a column vector containing the equalizer's setting at time instant  $k$  and  $^H$  denotes complex conjugate transposition. As the channel's transfer function is possibly non-minimum phase, its output statistics of order up to two contain only information about its amplitude whereas information about its phase is contained in higher-order statistics. This is why, on one hand, equalization is impossible (at the symbol rate) when the  $\{a_k\}$  have a Gaussian distribution and on the other hand statistics of order higher than two are required in order to identify the channel in a non-Gaussian case (which can be introduced indirectly via nonlinearities, as in the CMA algorithm).

The Constant Modulus Algorithm (CMA) [1][2] is one of the best known blind equalization algorithms and is a stochastic gradient algorithm for the cost function:

$$J_p(W) = \frac{1}{2p} E(|y|^p - 1)^2 \quad , \quad p \in \{1, 2, \dots\} \quad (1)$$

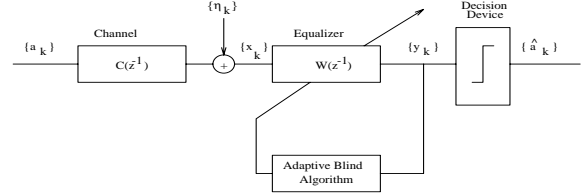


Figure 1. A typical blind equalization scheme

where  $E$  denotes statistical expectation. The corresponding stochastic gradient algorithm is given by:

$$W_{k+1} = W_k - \mu X_k y_k |y_k|^{p-2} (|y_k|^p - 1) \quad (2)$$

The well known CMA 2-2 algorithm is a special case of (2) for  $p = 2$  and has been shown to be able in general to open the communication system's eye. In this paper we will present a new class of algorithms based on the constant modulus property that impose a more severe constraining on the equalizer at each time instant.

## 2. MOTIVATION

In [3] we have derived a class of algorithms for blind equalization that minimize at each iteration the following deterministic criterion:

$$\| \text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_{k+1} \|_{P_k^{-1}}^2 + \left( \frac{1}{\mu} - 1 \right) \| W_{k+1} - W_k \|^2 \quad (3)$$

where  $\mathbf{X}_k = [X_k \ X_{k-1} \ \dots \ X_{k-L+1}]$ ,  $\|x\|_S^2 = x^H S x$ ,  $P_k = \mathbf{X}_k^H \mathbf{X}_k$  and the *sign* of a vector is defined as follows:

$$\text{sign}([a_1 \ a_2 \ \dots \ a_M]^T) = \left[ \frac{a_1}{|a_1|} \ \frac{a_2}{|a_2|} \ \dots \ \frac{a_M}{|a_M|} \right]^T \quad (4)$$

where  $^T$  denotes the transpose of a vector or a matrix. The corresponding algorithm (NSWCMA) is:

$$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k P_k^{-1} (\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k) \quad (5)$$

and can be seen as a blind-equalization counterpart of the Affine Projection Algorithm (APA) [4][5][6][7]. The first member of this class of algorithms ( $L = 1$ ) actually coincides with the recently proposed [8] Normalized Constant Modulus Algorithm (NCMA). However the APA suffers from a high steady-state error due to the influence of noise when both the parameters  $L$  and  $\bar{\mu}$  are chosen to be large. The main reason for this is the severe ill-conditioning at certain time instants of the sliding-window sample covariance matrix  $P_k$ . Indeed the noise gets amplified by  $P_k^{-1}$ . This

suggested some regularization of this sample covariance matrix. We propose to use a sample covariance matrix with an exponential instead of a rectangular window (in fact the use of an exponential window in a classical adaptive filtering context was first suggested in [5]). Simulations have shown that for a sample covariance matrix, the distribution of the condition number shows a mean that increases as the effective window length decreases (for both rectangular and exponential window). However this distribution has a much more pronounced tail for a rectangular window, compared to an exponential one (the rectangular window sample covariance matrix gets extremely ill-conditioned at certain time instants). We will show in the next section that the modified algorithm still results from a deterministic criterion.

### 3. DERIVATION

We first limit ourselves to a classical adaptive filtering context. We denote by  $d_k$  the desired response that has to be matched by the filter output at time instant  $k$  and by  $D_k$  a vector of the  $L$  most recent desired responses:

$$D_k = [d_k \ d_{k-1} \ \dots \ d_{k-L+1}]^T. \quad (6)$$

Now consider the following deterministic criterion:

$$\|D_k - \mathbf{X}_k^H W_{k+1}\|_{S_k}^2 + \|W_{k+1} - W_k\|^2, \quad (7)$$

where  $S_k = \mu^{-1} R_k - \mathbf{X}_k^H \mathbf{X}_k$  and  $R_k$  is an  $L \times L$  matrix updated as:  $R_k = \lambda R_{k-1} + \mathcal{X}_k \mathcal{X}_k^H$ , where  $\mathcal{X}_k$  is the first column of  $\mathbf{X}_k^H$ . It can be shown that this criterion is exactly minimized at each iteration by the following recursive algorithm:

$$\begin{aligned} R_k &= \lambda R_{k-1} + \mathcal{X}_k \mathcal{X}_k^H \\ W_{k+1} &= W_k + \mu \mathbf{X}_k R_k^{-1} (D_k - \mathbf{X}_k^H W_k). \end{aligned} \quad (8)$$

Eq. (8) describes a new parametric class of algorithms for adaptive filtering. There are three adjustable parameters:  $L$ ,  $\lambda$  and  $\mu$ .  $L$  is the number of constraints (see first term in (7)) imposed on the filter setting  $W_{k+1}$ ,  $\lambda$  controls the tracking and the conditioning of the  $L \times L$  sample covariance matrix  $R_k$  and  $\mu$  is a stepsize parameter that controls the deviation of the “next” filter  $W_{k+1}$  w.r.t. to  $W_k$ . In order to check the region of  $\mu$  that guarantees the stability of (8), we use the following asymptotic approach: by definition,  $S_k = \mu^{-1} R_k - \mathbf{X}_k^H \mathbf{X}_k$ . Taking expectation of both sides one obtains:

$$E(S_k) = R \left( -N + \frac{1}{\mu} \times \frac{1}{1-\lambda} \right), \quad (9)$$

where  $R$  is the true  $L \times L$  covariance matrix:  $R = E \mathcal{X}_k \mathcal{X}_k^H$ . Since the criterion (7) only makes sense for  $S_k$  positive definite,  $S_k$  should also be asymptotically positive definite, and thus the following must hold according to (9) :

$$\mu < \frac{1}{N(1-\lambda)}. \quad (10)$$

Eq. (10) shows that the choice of  $\mu$  must be done jointly with the choice of  $\lambda$  for a given length  $N$ . In order to get an idea of the convergence dynamics of (8), we examine the eigenvalues of the matrix  $I_N - \mu \mathbf{X}_k R_k^{-1} \mathbf{X}_k^H$ :

$$\begin{aligned} \mathbf{X}_k^H (I_N - \mu \mathbf{X}_k R_k^{-1} \mathbf{X}_k^H) &= (I_L - \mu \mathbf{X}_k^H \mathbf{X}_k R_k^{-1}) \mathbf{X}_k^H = \\ (R_k - \mu \mathbf{X}_k^H \mathbf{X}_k) R_k^{-1} \mathbf{X}_k^H &= \mu S_k R_k^{-1} \mathbf{X}_k^H. \end{aligned} \quad (11)$$

Eq. (11) shows that the eigenvalues  $\neq 1$  of  $I_N - \mu \mathbf{X}_k R_k^{-1} \mathbf{X}_k^H$  are also the eigenvalues of  $\mu S_k R_k^{-1}$ . The matrix  $\mu S_k R_k^{-1}$  is asymptotically equal to:

$$\begin{aligned} \mu (E S_k) (E R_k)^{-1} &= \mu (1-\lambda) \left[ \frac{1}{\mu(1-\lambda)} - N \right] I_L \\ &= (1 - \mu N (1-\lambda)) I_L. \end{aligned} \quad (12)$$

Eq. (12) shows that the speed of convergence of the algorithm increases as the quantity  $\mu N (1-\lambda)$  increases, provided of course that (10) is verified and therefore  $\mu N (1-\lambda)$  plays the role of an *effective stepsize*. This gives us enough degrees of freedom to choose the algorithm’s parameters so as to provide a high convergence speed.

For constant-modulus blind equalization, we have shown in [3] that the optimization problem:

$$\begin{aligned} \min \|W_{k+1} - W_k\|^2 \\ \text{subject to: } |X_{k-i}^H W_{k+1}| = 1, \quad i = 0, 1, \dots, L-1, \end{aligned} \quad (13)$$

leads to a classical adaptive filtering problem with desired response:

$$D_k = \text{sign}(\mathbf{X}_k^H W_k). \quad (14)$$

This has allowed us to formulate the following *separation principle*: generate a desired response by projecting the *a priori* equalizer output on the unit circle and then run any classical adaptive filtering algorithm with that desired response. So the proposed class of algorithms corresponds to (7) and (8) with the desired response vector  $D_k$  chosen as in (14). When  $\|S_k\| \rightarrow 0$ , then the criterion (7) with  $D_k$  as in (14) reduces to (13).

### A Decision-Directed version

If one replaces in (7), (8), the vector  $D_k = \text{sign}(\mathbf{X}_k^H W_k)$  by the vector  $\hat{A}_k$  defined as:

$$\hat{A}_k = [\hat{a}_{k,0} \ \hat{a}_{k,1} \ \dots \ \hat{a}_{k,L-1}]^T, \quad (15)$$

where  $\hat{a}_{k,i} = \text{decision}(X_{k-i}^H W_k)$ , one obtains a decision-directed (DD) version of (8). When used with constant-modulus (CM) constellations, our simulations have shown that this DD version converges as well and at about the same speed as its CMA counterpart corresponding to ((8), (14)). Furthermore, the DD version has a lower steady-state error than its CMA counterpart. So the DD version appears to be preferable for CM constellations. For non-constant-modulus QAM constellations, however, the DD algorithm does not seem to converge, similarly to what has been found for PAM constellations in the 70’s. However, a modified DD scheme can be proposed. We propose to classify the equalizer’s output at the decision device not by finding its closest symbol of the *transmitted* constellation but of a constant modulus constellation obtained by projecting all the transmitted constellation’s symbols on a circle in such a way so as to maintain each symbol’s angle. The version of (8) thus derived corresponds to a “DD” algorithm that seems to work well under severe Inter-Symbol-Interference (ISI) even for non-constant-modulus constellations! These aspects are the subject of ongoing research.

### 4. ALGORITHMIC ORGANIZATION

A low-complexity algorithmic organization for the Affine Projection Algorithm has been proposed in [9]. We will now give a similar algorithmic organization for the algorithm (8). We denote by  $E_k$  and by  $\mathcal{E}_{k-1}$  the algorithm’s *a priori* and *a posteriori* errors at time instants  $k$  and  $k-1$ , respectively:

$$\begin{aligned} E_k &= D_k - \mathbf{X}_k^H W_k \\ \mathcal{E}_{k-1} &= D_{k-1} - \mathbf{X}_{k-1}^H W_k. \end{aligned} \quad (16)$$

The fast algorithm derived in [9] for the APA is based (among others) on the following simple relationship between *a priori* and *a posteriori* errors:

$$E_k = \begin{bmatrix} E_{k,1} \\ (1-\mu)\bar{\mathcal{E}}_{k-1} \end{bmatrix}, \quad (17)$$

where  $E_{k,i}$  denotes the  $i^{\text{th}}$  element of the vector  $E_k$  and  $\bar{\mathcal{E}}_{k-1}$  denotes the  $(L-1) \times 1$  vector of the  $L-1$  lower-most elements of  $\mathcal{E}_{k-1}$  (same for  $\bar{E}_{k-1}$ ). This relation has allowed in [9] for a very efficient computation of  $E_k$ , which is needed at each step of the algorithm. In the case of the algorithm (8) however, the corresponding relationship is:

$$E_k = \begin{bmatrix} E_{k,1} \\ \bar{E}_{k-1} \end{bmatrix} = \begin{bmatrix} E_{k,1} \\ 0_{(L-1) \times 1} \end{bmatrix} + Z(I_L - \mu P_{k-1} R_{k-1}^{-1}) E_{k-1}, \quad (18)$$

where  $Z$  is a  $L \times L$  matrix with 1's on its first subdiagonal and zeros elsewhere. Apart from the identities related to the updating of  $E_k$ , the rest of the algorithmic organization presented in [9] for the APA is still applicable to the algorithm (8). Before presenting the resulting algorithm, we first introduce some notation:

$\alpha_k, \beta_k, E_{\alpha,k}, E_{\beta,k}$  denote the forward and backward linear predictors and the corresponding prediction error energies of the sample covariance matrix  $R_k$ , respectively, and are needed for the updating of  $R_k^{-1}$  by a FTF algorithm.  $C_k$  denotes the direct Kalman gain vector of  $R_k^{-1}$  defined as  $C_k = R_k^{-1} \mathcal{X}_k$  and  $\gamma_k$  the inverse of the so-called "likelihood variable" ( $\gamma_k = \frac{1}{1 - \mathcal{X}_k^H C_k}$ ).  $l_k$  is the quantity  $R_k^{-1} E_k$  and  $F_k$  is a  $L \times 1$  vector defined as:

$$F_k = \begin{bmatrix} E_{k,1} \\ E_{k,2} + E_{k-1,1} \\ \vdots \\ E_{k,L} + E_{k-1,L-1} + \dots + E_{k-L+1,1} \end{bmatrix}, \quad (19)$$

and is an essential quantity for the algorithm's updating.  $\widehat{W}_{k+1}$  is an intermediate quantity that can be updated with a gradient proportional to only one input data vector (see(21)) and is defined as:

$$\widehat{W}_{k+1} = W_{k+1} - \mu \bar{\mathbf{X}}_k \bar{F}_k, \quad (20)$$

where  $\bar{\mathbf{X}}_k$  denotes the  $N \times (L-1)$  submatrix of  $\mathbf{X}_k$  that consists of its  $L-1$  left-most columns. We also denote by  $\tilde{\mathcal{X}}_k$  the vector of the  $L-1$  lower-most elements of  $\mathcal{X}_k$  and by  $\bar{F}_k$  the vector of the  $L-1$  upper-most elements of  $F_k$ . The key recursive formula of the algorithm is the following identity:

$$\widehat{W}_{k+1} = \widehat{W}_k + \mu X_{k-L+1} F_{k,L}. \quad (21)$$

Finally the quantity  $\tau_k$  is defined as  $\tau_k = X_k^H \bar{\mathbf{X}}_{k-1}$ , where  $\bar{\mathbf{X}}_{k-1}$  contains the  $L-1$  right-most columns of  $\mathbf{X}_k$ . We now give the equations of the resulting efficient algorithm that realizes (8):

0. Initialization:  $\alpha_0 = [1 \ 0_{(L-1) \times 1}]^T, \beta_0 = [0_{(L-1) \times 1} \ 1]^T, E_{\alpha,0} = E_{\beta,0} = \delta$  (a small positive number).

1. Use a (prewindowed) Stabilized Fast Transversal Filter algorithm [10] (prediction part) to update  $E_{\alpha,k}, E_{\beta,k}, \alpha_k, \beta_k, C_k$  and  $\gamma_k$ .

2.  $\tau_k = \tau_{k-1} + x_k \tilde{\mathcal{X}}_k^H - x_{k-N} \tilde{\mathcal{X}}_{k-N}^H$

3.  $\widehat{E}_{k,1} = d_k - X_k^H \widehat{W}_k$

4.  $E_{k,1} = \widehat{E}_{k,1} - \mu \tau_k \bar{F}_{k-1}$

5.  $\Omega_k = \mu R_{k-1}^{-1} E_{k-1}$

6.  $E_k = \begin{bmatrix} E_{k,1} \\ 0_{(L-1) \times 1} \end{bmatrix} + Z(E_{k-1} - P_{k-1} \Omega_k)$

7.  $R_k^{-1} = \frac{1}{\lambda} R_{k-1}^{-1} - \gamma_k C_k C_k^H$

8.  $P_k = P_{k-1} - \mathcal{X}_{k-N} \mathcal{X}_{k-N}^H + \mathcal{X}_k \mathcal{X}_k^H$

9.  $l_k = R_k^{-1} E_k$

10.  $F_k = \begin{bmatrix} 0 \\ \bar{F}_{k-1} \end{bmatrix} + l_k$

11.  $\widehat{W}_{k+1} = \widehat{W}_k + \mu X_{k-L+1} F_{k,L}$

If a prewindowed Stabilized FTF is used in step 1, this will take  $6L$  multiplications. The corresponding computational complexity for the other steps in terms of multiplications is:  $L$  for each of steps 2 and 4,  $N$  for each of steps 3 and 11,  $L^2$  for each of steps 6,8,9,  $2L^2 + L$  for step 5 and  $L^2 + L$  for step 7. This gives an overall complexity of  $2N + 6L^2 + 10L$  for an algorithm that corresponds exactly to the criterion (8). The  $O(L^2)$  term represents the price paid for implementing exactly an algorithm that uses a regularized covariance matrix, without any approximation. On the contrary, the FAP in [9] has a complexity of  $2N + 20L$  (numerically stable version) but corresponds to an algorithm with a non-regularized sample covariance matrix  $R_k = \mathbf{X}_k^H \mathbf{X}_k$  and is only approximative when some sort of regularization is introduced (it is similarly possible however to achieve a lower complexity for (8) if approximations are introduced). However, as usually  $L$  is chosen to be significantly smaller than  $N$  (especially in acoustic echo cancellation problems), our algorithm's complexity is still comparable to that of FAP. For example, in a blind equalization case with  $N = 30$  and  $L = 5$ , our algorithm will have a complexity of 260 multiplications/iteration whereas FAP has a complexity of 160 multiplications/iteration.

The algorithmic organization proposed above for the general adaptive-filtering case is still applicable in the blind-equalization context, but in this case the desired signal vector  $D_k$  should rather be defined as follows:

$$D_k = [\text{sign}(X_k^H W_k) \cdots \text{sign}(X_{k-L+1}^H W_{k-L+1})]^T, \quad (22)$$

instead of  $D_k = \text{sign}(\mathbf{X}_k^H W_k)$ , in order to have (18) satisfied. However this approximation has been found in computer simulations not to have a considerable impact on performance.

## 5. COMPUTER SIMULATIONS

In a first blind-equalization experiment a 2-PAM random signal is transmitted through a MA(2) channel with impulse response  $h_c = [1 \ 2 \ 0.6]$  and a white Gaussian noise  $\eta_k$  is added to the received signal resulting to an SNR of 20 dB. An adaptive blind algorithm as shown in figure 1 is used to update the equalizer. As a measure of performance we use the system's closed-eye measure defined as:

$$\rho = \frac{\sum_i |s_i| - \max_i |s_i|}{\max_i |s_i|}, \quad (23)$$

where  $s$  represents the overall communication system's impulse response. The eye is said to be open when  $\rho < 1$  and closed otherwise. Figure (2a) shows the evolution of  $\rho$  averaged over 100 Monte-Carlo simulations for four different blind equalization algorithms. The CMA 2-2 is employed with a stepsize  $\mu = 0.04$ , the NCMA (NSWCMA

with ( $L = 1$ ) with  $\bar{\mu} = 0.3$ , the algorithm proposed in [11] (that we call RLS-CMA) with a forgetting factor  $\lambda = 0.94$  and our proposed algorithm ((8),(14)) with  $L = 6$ ,  $\lambda = 0.5$  and  $\mu = 0.01$ . The equalizer's length is equal to 6 and all algorithms are initialized with  $W_0 = 10^{-4} \times [1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$ . It can be seen that our proposed algorithm has an increased convergence speed and opens the channel's eye faster than the other algorithms. It is also noted that it behaves well in steady-state, despite the small value for the forgetting factor it uses. This means that even such a small value regularizes adequately the received signal's sample covariance matrix. The algorithm has been found to have a similar behaviour when it was tried on 4-QAM and 16-QAM constellations.

Figure (2b) shows a case where the problem of false minima is encountered. The transmission channel's impulse response is now  $h_c = [1 \ 0.6 \ 0.36]$  and a 3-tap equalizer is used, initialized at  $[0 \ 0 \ 1]^T$ . This is a typical example where the CMA's get trapped by a local minimum of their cost function, being unable to open the system's eye. The RLS-CMA of [11] is used with a forgetting factor  $\lambda = 0.95$  and then our algorithm ((8),(14)) with  $L = 3$ ,  $\lambda = 0.95$  and  $\mu = 1$ . It can be seen how the first algorithm is indeed trapped by a local minimum and does not open the eye, whereas the second one is able of converging to a setting that opens the eye. This fact reflects the potential advantage of the algorithms of the proposed class that permit the use of a big effective stepsize that guarantees stability on one hand but also provides a large enough movement around false minima that helps escaping from them (a discussion about this potential ability of normalized CMA's can be found in [12]).

## 6. CONCLUSIONS

We have proposed a new class of adaptive filtering algorithms for blind equalization of constant modulus signals. Our motivation has been the high steady-state MSE sometimes present in algorithms of the NSW-CMA class (5) when big values for  $\bar{\mu}$  and for  $L$  are used and the proposed remedy has been to use a more regularized sample-covariance matrix using exponential forgetting as shown in (8). The algorithms have been derived by minimizing exactly at each iteration a deterministic criterion, both in the blind-equalization and in the classical-adaptive-filtering context and an asymptotic analysis has revealed the role of an *effective stepsize* and has provided parameter bounds that guarantee stability. A decision-directed version was also proposed. Based on an algorithm given in [9] we propose an algorithmic organization of complexity  $2N + 6L^2 + 10L$  multiplications/iteration. The algorithms' behaviour has been tested via computer simulations that show their increased convergence speed w.r.t. other constant-modulus algorithms as well as their ability to escape from false minima of their cost function.

## REFERENCES

- [1] D. N. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communications Systems," *IEEE Trans. Commun.*, vol. COM-28, pp. 1867-1875, Nov. 1980.
- [2] J. R. Treichler and B. G. Agee, "A New Approach to Multipath Correction of Constant Modulus Signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 459-472, Apr. 1983.
- [3] C. B. Papadias and D. T. M. Slock, "Normalized Sliding Window Constant Modulus Algorithms for Blind Equalization", *Proc. 14<sup>th</sup> GRETSI Symposium on Signal and Image Processing*, Juan les Pins, France, Sept. 13-16, 1993.

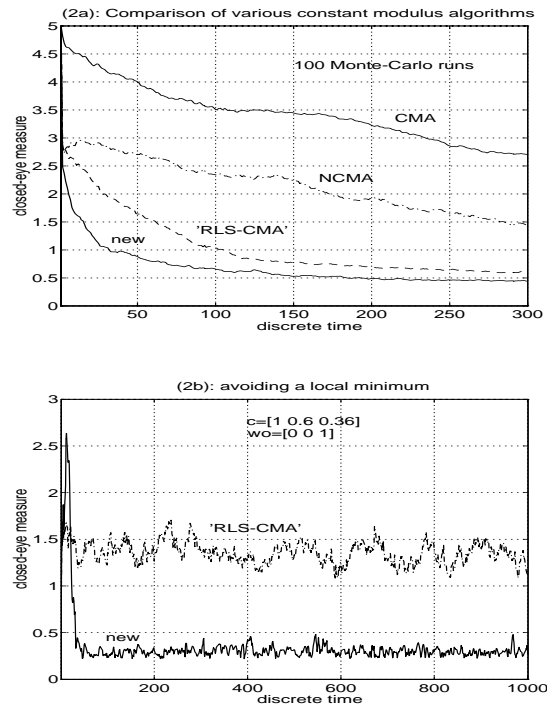


Figure 2. Computer simulations results

- [4] K. Ozeki, T. Umeda, "An Adaptive Filtering Algorithm Using an Orthogonal Projection to an Affine Subspace and its Properties", *Electronics and Communications in Japan*, vol. 67-A, No. 5, 1984.
- [5] D. T. M. Slock, "Underdetermined Growing and Sliding Window Covariance Fast Transversal Filter RLS Algorithms," *Proc. EUSIPCO 92, Vth European Signal Processing Conference*, pp. 1169-1172, Brussels, Belgium, Aug. 24-27, 1992.
- [6] D. T. M. Slock, "The Block Underdetermined Covariance (BUC) Fast Transversal Filter (FTF) Algorithm for Adaptive Filtering," *Proc. Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 26-28, 1992.
- [7] M. Montazeri and P. Duhamel, "A Set of Algorithms Linking NLMS and Block RLS Algorithms," submitted, *IEEE Trans. on Signal Processing*.
- [8] K. Hilal and P. Duhamel, "A Convergence Study of the Constant Modulus Algorithm Leading to a Normalized-CMA and a Block-Normalized-CMA," *Proc. EUSIPCO 92, Vth European Signal Processing Conference*, pages 135-138, Brussels, Belgium, Aug. 24-27, 1992.
- [9] S. L. Gay, "A Fast Converging, Low Complexity Adaptive Filtering Algorithm," *Proc. Third International Workshop on Acoustic Echo Control*, pp. 223-226, Plestin les Grèves, France, Sept. 7-8, 1993.
- [10] D. T. M. Slock and T. Kailath, "Numerically Stable Fast Transversal Filters for Recursive Least-Squares Adaptive Filtering," *IEEE Trans. on Signal Processing*, vol. ASSP-39, pp. 92-114, Jan. 1991.
- [11] R. Gooch, M. Ready, and J. Svoboda, "A Lattice-Based Constant Modulus Adaptive Filter," *Proc. Asilomar Conference on Signals, Systems and Computers*, pp. 282-286, Pacific Grove, CA, 1987.
- [12] C. B. Papadias and D. T. M. Slock, "On the Convergence of Normalized Constant Modulus Algorithms for Blind Equalization," *Proc. DSP International Conference on Digital Signal Processing*, Nicosia, Cyprus, July 14-16, 1993.