AN ACCELERATED CONSTANT MODULUS ALGORITHM FOR SPACE-TIME BLIND EQUALIZATION

Magno T. M. Silva[1]*, Max Gerken*[†]*, and Maria D. Miranda*[2]

[1]University of São Paulo - EPUSP - PTC
Av. Prof. Luciano Gualberto, 158 , trav. 3
05508-900 São Paulo, SP, Brazil
phone: +55 11 3091 5606, fax: +55 11 3091 5128
email: magno@lcs.poli.usp.br

[2]Mackenzie University- PPGEE
Rua da Consolação, 869
01302-907 São Paulo, SP, Brasil
phone: +55 11 3236 8711, fax: +55 11 3091 5128
email: mdm@mackenzie.com.br

ABSTRACT

We address the problem of signal separation using space-time blind equalization techniques. A novel blind algorithm, denoted ACMA (Accelerated Constant Modulus Algorithm), is proposed. It minimizes the constant modulus cost function and is based on a tuner used in adaptive control that sets the second derivative ("acceleration") of the coefficient estimates. Both the convergence speed and computational complexity of Multiuser ACMA lie between those of the Multiuser Constant Modulus and the Multiuser Shalvi-Weinstein algorithms. Some preliminary results show that the proposed algorithm presents a robust behavior with respect to global minima and tracking capability.

1. INTRODUCTION

Nowadays, efficient equalizers have had an important role in mobile communications. A typical problem that frequently arises in multiuser communication systems is the blind separation of linear mixtures of signals. In this context, several different algorithms have been proposed. The Multiuser Constant Modulus Algorithm (MU-CMA) [1] and the Multiuser Shalvi-Weinstein Algorithm (MU-SWA) [2] are based on a stochastic gradient approach. They present an unfavourable compromise between convergence speed and computational complexity. Recently the Quasi-Newton Cross-Correlation Constant Modulus Algorithm was proposed in [3]. Compared to MU-CMA, it presents faster convergence rate, but higher complexity. Moreover, it is subject to numerical instability [3, 2]. In this scenario, designing a stable algorithm which has a more favorable compromise between efficient behavior and computational complexity is a problem of interest.

A discrete-time adaptive filtering algorithm was introduced in [4] and further analyzed in [5]. It was derived from a continuous-time tuner used in adaptive control, which sets the second derivative ("acceleration") of the coefficient estimates [6]. For this reason the discrete-time algorithm was named the Accelerating Adaptive Filtering (AAF) algorithm. At the cost of a moderate increase in computational complexity, this algorithm shows some advantages when compared to the LMS (Least Mean Squares) and NLMS (Normalized LMS) algorithms. In [4] it was shown that, for colored input signals, AAF presents a more favorable compromise between convergence speed and steady-state estimation error than LMS or NLMS.

Inspired on the general methodology for the design of blind adaptive algorithms proposed in [7], we derive an algorithm for space-time blind equalization based on the AAF algorithm. Since AAF compares favorably to LMS, one may expect that the resulted algorithm would achieve better performance than MU-CMA.

In the next section problem formulation is presented and space-time blind algorithms are revisited. This is followed by a summary of the continuous-time accelerating tuner and the associated Multiuser Accelerated Constant Modulus Algorithm (MU-ACMA). Then we show simulation results comparing the convergence and tracking behavior of ACMA, CMA and SWA for SISO (single-input single-output) and MIMO (multiple-input multiple-output) systems. A conclusion section closes the paper.

This work was supported by [1]FAPESP (00/12350-6) and [2]Mackpesquisa.

2. ISSUES ON SPACE-TIME BLIND EQUALIZATION

Let a MIMO system with N sources and with an antenna array which has $L > N$ sensors as depicted in Fig. 1. The source sequences $a_i(n)$, $i = 1,\ldots,N$ are assumed i.i.d., independent from one another, non Gaussian, and zero-mean. The transmitted signals suffer inter-symbol and co-channel interferences. The channel from the i^{th} source to the j^{th} sensor is modelled by an FIR filter with K_c coefficients and η_i, $i = 1,2,\ldots,L$ represent additive white Gaussian noise. The outputs of the L sensors are processed with N parallel space-time FIR equalizers, each with K_t time diversity and $M = LK_t$ taps. The blind equalizer must mitigate the channel effects without the data training. The i^{th} equalizer's output can be written as $y_i(n) = \mathbf{w}_i^T(n-1)\mathbf{u}(n)$, where $\mathbf{u}(n)$ and $\mathbf{w}_i(n-1)$ are the input and the weight equalizer vectors, respectively.

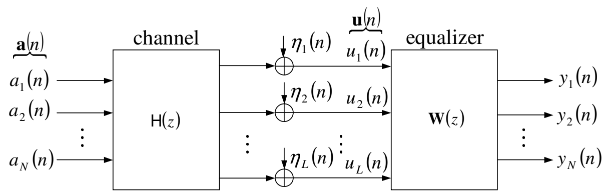

Figure 1: MIMO equivalent system model.

In the case of joint blind simultaneous recovery of all input signals, the Godard cost function is given by [1]

$$J_G = \sum_{i=1}^{N}\left[J_{Gi} + \frac{\xi}{2}\sum_{j=1,j\neq i}^{N}\sum_{\delta=-\delta_1}^{\delta_1}|r_{ij}(\delta)|^2\right] \quad (1)$$

in which $J_{Gi} = \mathrm{E}\{(|y_i(n)|^2 - R_2^a)^2\}$, $R_2^a = \mathrm{E}\{|a(n)|^4\}/\mathrm{E}\{|a(n)|^2\}$, $r_{ij}(\delta) = \mathrm{E}\{y_i(n)y_j^*(n-\delta)\}$, $\delta_1 = K_t + K_c - 1$, and $*$ stands for complex conjugate. Note that the sources are assumed with the same statistics. The second term of the right side of (1) is introduced to penalize the cross-correlations between different users through weight $\xi/2$ [1]. The gradient vector of this cost function related to the i^{th} user is given by

$$\boldsymbol{\nabla}_i J_G = \mathrm{E}\left\{e_i(n)\mathbf{u}^*(n)\right\} + \frac{\xi}{2}\sum_{i=1,i\neq i}^{N}\sum_{\delta=-\delta_1}^{\delta_1}\mathrm{E}\left\{y_j(n)\mathbf{u}^*(n)\right\}r_{ij}(\delta) \quad (2)$$

where $$e_i(n) = (|y_i(n)| - R_2^a) y_i(n). \quad (3)$$

It is usual to estimate the cross-correlation $r_{ij}(\delta)$ with an exponential window, considering a forgetting factor λ, and the other expectations with instantaneous estimates [1, 3]. With these estimates, "stochastic gradient algorithms" can be characterized by the following equations

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) - \mu(n)\check{e}_i(n)\mathbf{u}^*(n) \quad (4)$$

in which

$$\check{e}_i(n) = e_i(n) + \rho_i(n) \quad \text{and} \quad (5)$$

$$\rho_i(n) = \frac{\xi}{2} \sum_{j=1, j\neq i}^{N} \sum_{\delta=-\delta_1}^{\delta_1} y_j(n)\hat{r}_{ij}(\delta). \quad (6)$$

In MU-CMA [1], the adaptation step is a constant scalar $\mu(n) = \mu$. In MU-SWA, $\mu(n) = -\gamma^{-1}\mathbf{R}^{-1}(n)$, being $\mathbf{R}(n)$ an estimate of the autocorrelation matrix $\mathrm{E}\{\mathbf{u}^T(n)\mathbf{u}^*(n)\}$, $\gamma = R_2^a - \beta\mathrm{E}\{|a(n)|^2\}$, and $\beta = 2 (= 3)$ in the complex (real) case. This algorithm was recently proposed in [2] as an approximation of the higher-order cumulant-based algorithm [8] with capacity to simultaneously recover all source sequences and can be interpreted as an extension of SWA [9] for the multiuser environment. It is possible to adjust the parameters of MU-SWA and MU-CMA to reach the same steady-state mean-square error (MSE) by using [2]

$$\mu = -(1-\lambda)/(\gamma\sigma_u^2) \quad (7)$$

where σ_u^2 is the variance of the input signal and λ the forgetting factor used to recursively update the estimate $\mathbf{R}^{-1}(n)$.

3. THE ACCELERATING TUNER

In adaptive control and recursive coefficient estimation one often needs to adjust recursively an estimate $\mathbf{w}(t)$ of a coefficient vector $\mathbf{w}_o$ using a measured signal $d(t) = \mathbf{u}^T(t)\mathbf{w}_o + \eta(t)$, where $\mathbf{u}(t)$ is the input signal vector and $\eta(t)$ the measurement noise. The goal is to maintain both the estimation error $\varepsilon(t) = \mathbf{u}^T(t)\mathbf{w}(t) - d(t)$ and the coefficient error $\widetilde{\mathbf{w}}(t) = \mathbf{w}(t) - \mathbf{w}_o$ as small as possible.

The most straightforward tuning method used in adaptive control sets the first derivative ("velocity") of the coefficient estimates proportional to the estimation error:

$$\dot{\mathbf{w}}(t) = -\mathbf{M}\mathbf{u}^*(t)\varepsilon(t) \quad (8)$$

$$\varepsilon(t) = \mathbf{u}^T(t)\mathbf{w}(t) - d(t) \quad (9)$$

where $\mathbf{M}$ is a positive-definite matrix of dimensions $M \times M$.

A tuner that adjusts the second derivative ("acceleration") of the coefficient estimates was introduced in [6]. Observing that $\dot{\widetilde{\mathbf{w}}}(t) = \dot{\mathbf{w}}(t)$ and defining $\mathbf{q}(t) = \dot{\widetilde{\mathbf{w}}}(t)$, the accelerating tuner can be described as follows:

$$\dot{\mathbf{w}}(t) = \mathbf{q}(t) \quad (10)$$

$$\dot{\mathbf{q}}(t) = -\mathbf{M}_1\mathbf{u}^*(t)\varepsilon(t) - 2\mathbf{M}_1\left(\mathbf{M}_2 + \mathbf{u}^*(t)\mathbf{u}^T(t)\mathbf{M}_1\mathbf{M}_3\right)\mathbf{q}(t) \quad (11)$$

$$\varepsilon(t) = \mathbf{u}^T(t)\mathbf{w}(t) - d(t) \quad (12)$$

where the $M \times M$ symmetric matrices $\mathbf{M}_k$, $k = 1, 2, 3$ are positive-definite. If measurement noise is absent we can write $d(t) = \mathbf{u}^T(t)\mathbf{w}_o$ and $\varepsilon(t) = \mathbf{u}^T(t)\widetilde{\mathbf{w}}(t)$. In this case the dynamics of the accelerating tuner can be described by using the coefficient error vector $\widetilde{\mathbf{w}}(t)$:

$$\underbrace{\begin{bmatrix} \dot{\widetilde{\mathbf{w}}}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}}_{\dot{\mathbf{v}}(t)} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}}_{\mathbf{A}(t)} \underbrace{\begin{bmatrix} \widetilde{\mathbf{w}}(t) \\ \mathbf{q}(t) \end{bmatrix}}_{\mathbf{v}(t)}, \quad \text{with}$$

$$\mathbf{A}_{21} = -\mathbf{M}_1\mathbf{u}^*(t)\mathbf{u}^T(t)$$

$$\mathbf{A}_{22} = -2\mathbf{M}_1\left(\mathbf{M}_2 + \mathbf{u}^*(t)\mathbf{u}^T(t)\mathbf{M}_1\mathbf{M}_3\right).$$

Sufficient conditions to ensure stability of this system, established by a Lyapunov function defined in [6], are

$$4\mathbf{M}_1\mathbf{M}_3\mathbf{M}_1\mathbf{M}_2 > \mathbf{I} \quad \text{and} \quad (13)$$

$$\mathbf{M}_2\mathbf{M}_1\mathbf{M}_3 + \mathbf{M}_1\mathbf{M}_3\mathbf{M}_2 > \mathbf{M}_1^{-1}/2. \quad (14)$$

Among different methods to obtain a discrete-time algorithm from a continuous-time one, the direct and reverse Euler methods are the simplest. Considering the first order differential equation $\dot{f}(t) = g(t)$, its discrete-time version according to direct Euler method is $f(n+1) = f(n) + Tg(n)$ and to reverse Euler method is $f(n) = f(n-1) + Tg(n)$, where T is the integration constant. It is relevant to note that the popular LMS algorithm can be obtained by applying direct Euler method to discretize the previously "velocity" tuner making $\mathbf{M} = \mu_o\mathbf{I}$ and $\mu = \mu_o T$.

The AAF algorithm was obtained [4] using (10)-(12) by Euler's reverse method. The reasons for this choice are quite simple: the direct Euler method results in a low complexity algorithm that may be unstable; other numerical integration methods like the trapezoidal rule result in algorithms with higher computational complexity. Thus, we also use Euler's reverse method to derive ACMA in the next section.

4. ACCELERATED BLIND ALGORITHMS

Replacing (9) by a nonlinear version based on the Godard cost function [10], i.e.

$$\varepsilon(t) = \varphi\big(y_c(t)\big) = (|y_c(t)|^2 - R_2^a)y_c(t) \quad (15)$$

being $y_c(t) = \mathbf{u}^T(t)\mathbf{w}(t)$ and applying the direct Euler method to discretize (8) and (15) by setting $\mathbf{M} = \mu_o\mathbf{I}$ and $\mu = \mu_o T$, we obtain the well-known Constant Modulus Algorithm.

Now suppose that we wish to extract only the i^{th} source and that the co-channel interference is absent, which is equivalent to the SISO case. ACMA is obtained by applying the reverse Euler method to (10), (11) and (15) with the integration step μ and the *a priori* error (3). Moreover, by using a linear approximation of $\varphi(\cdot)$ and making $\mathbf{M}_k = m_k\mathbf{I}$, $k = 1, 2, 3$, being m_k positive constants and $\mathbf{I}$ the identity matrix, we obtain the low complexity version of ACMA of Table 1. Note that α, $1/\alpha$, $\mu m_1\alpha$, $2\mu m_1^2 m_3$ and $\mu^2 m_1$ need to be computed only once. A detailed derivation of this algorithm is shown in Appendix.

Initialize the algorithm by setting:
$R_2^a = \mathrm{E}\{\lvert a(n)\rvert^4\}/\mathrm{E}\{\lvert a(n)\rvert^2\}$
$\mathbf{w}_i(0) = [0 \ \cdots \ 0 \ 1 \ 0 \cdots \ 0]^T$, $\mathbf{q}_i(0) = \mathbf{0}$
$\alpha = 1 + 2\mu m_1 m_2$
$\beta = 2 \ (= 3)$ for the complex (real) case
For each instant of time, $n = 1, 2, \cdots$, compute:
$y_i(n) = \mathbf{u}^T(n)\mathbf{w}_i(n-1)$
$e_i(n) = (\lvert y_i(n)\rvert^2 - R_2^a)y_i(n)$
$b_i(n) = 2\mu m_1^2 m_3 + \mu^2 m_1(\beta\lvert y_i(n)\rvert^2 - R_2^a)$
$c_i(n) = \dfrac{b_i(n)\mathbf{u}^T(n)\mathbf{q}_i(n-1) + \mu m_1\alpha e_i(n)}{\alpha + b_i(n)\,\lVert\mathbf{u}(n)\rVert^2}$
$\mathbf{q}_i(n) = \dfrac{1}{\alpha}\left[\mathbf{q}_i(n-1) - c_i(n)\mathbf{u}^*(n)\right]$
$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mu\mathbf{q}_i(n)$

Table 1: Summary of ACMA.

The first problem we face when using ACMA is how to choose the parameters μ, m_1, m_2, and m_3 to obtain adequate performance. Motivated by the fact that the AAF algorithm is stable and reaches its fastest convergence at the upper bound of (13) [5] we introduce a parameter κ to set $m_1 m_2 = 1/(2\kappa)$ and $m_1 m_3 = \kappa/2$. This guarantees $4m_1^2 m_2 m_3 = 1$. Consequently, we need to set three positive parameters, μ, κ and m_1, to adjust the performance of ACMA. However, our experience indicates that this is not a critical matter. Simulations results suggest that by choosing $m_1 \approx 0.15$ and $\kappa \approx 100$, we should vary the step size μ in the interval $]0,1[$ to obtain adequate algorithm performance.

Now considering the simultaneous recovery of all sources in presence of co-channel interference, MU-ACMA can be obtained by replacing $e_i(n)$ in Table 1 by $\breve{e}_i(n)$ defined in (5).

Table 2 shows the computational complexity of the algorithms for real signals. MU-ACMA has a computational complexity that lies between those of MU-CMA and MU-SWA, maintaining the proportionality to M operations per iteration.

Op.	MU-SWA	MU-ACMA	MU-CMA
$\times$	$4M^2+M(2N+$ $+1)+N(3D+5)$	$M(5N+1)+$ $N(3D+8)$	$M(2N)+$ $+N(3D+5)$
$\div$	1	N	–

Table 2: Computational complexity of the algorithms for real signals, $D{=}(N-1)(2\delta_1+1)$, $N \geq 2$.

5. SIMULATION RESULTS

In this section we compare the performance of ACMA to CMA and SWA, presenting results for the SISO and MIMO cases. The algorithms are adjusted to reach the same steady-state MSE using (7) for CMA and SWA and an experimental inspection for ACMA.

The contour plot of the SISO Godard cost function for the channel $H(z) = [1+0.6z^{-1}]^{-1}$ is shown in Fig. 2. This figure shows two local and two global minima of the cost function and different trajectories of CMA, SWA and ACMA. If initialized at $\mathbf{w}(0) = [0\ \ 1]^T$ the algorithms present similar behavior. For $\mathbf{w}(0) = [-0.4\ \ 0.05]^T$, which is close to one of the local minima, ACMA crosses the local minimum and reaches one of the global minima while CMA and SWA stagnate at the local minimum. If the pole of the channel is changed from -0.6 to -0.2, the local minima become more pronounced. In this case ACMA shows the same behavior of CMA and SWA, being attracted to the local minima. This behavior suggests that, to some extent, ACMA has the ability to escape from soft local minima. For $\mathbf{w}(0) = [0.2\ \ -0.4]^T$ ACMA reaches one of the global minima crossing the attraction domain of a local minimum while CMA and SWA converge to the other global minimum. In this case, ACMA follows the "rocky trail" while the others converge straight to a valley. While successfully avoiding a local minimum, ACMA shows a slower convergence rate. If the pole of the channel is again changed from -0.6 to -0.2, the three algorithms present similar behavior, converging to the same global minimum. Although the results of Fig. 2 are very interesting, we can conclude that ACMA does not avoid deep local minima.

We now consider a MIMO system with $N = 2$ users, $L = 3$ sensors and time-varying channels $H_{ij}(z) = h_0^{ij}(n) + h_1^{ij}(n)z^{-1} + h_2^{ij}(n)z^{-2}$, $i = 1,\ldots,N$, $j = 1,\ldots,L$, whose coefficients are generated by passing Gaussian white noise through a second order Butterworth filter designed to simulate a fade rate of 0.1 Hz [11]. Moreover, it is assumed 2-PAM modulation, SNR=30 dB, and 2 equalizers with 15 taps initialized with only two non-null elements at fifth and seventh positions respectively. Fig. 3 shows the equalizer-1's output error for MU-CMA, MU-ACMA, and MU-SWA. The error bursts can be associated with rapid changes of the channels' roots and deep spectral nulls. MU-SWA shows the faster convergence followed by MU-ACMA. In this case, MU-CMA shows the worst tracking capability. The corresponding residual interference (RI) curves are presented in Fig. 3-d. Note that the error bursts occur when the residual interference is above -10 dB. For the equalizer-2 the algorithms show similar behavior to the observed one in Fig. 3.

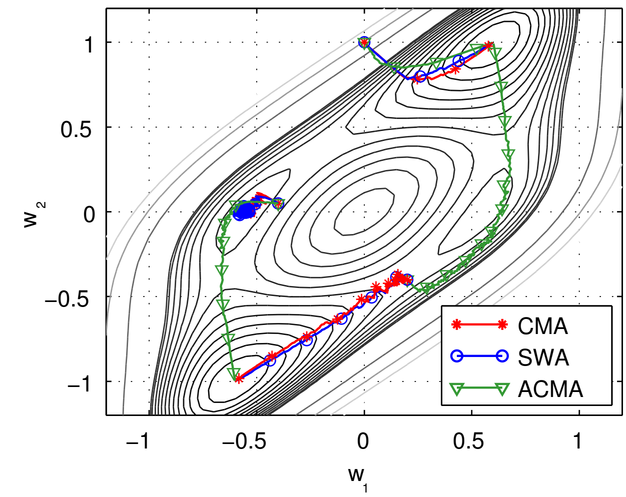

Figure 2: Contour of Godard cost function for 2-PAM related to the channel $H(z) = [1+0.6z^{-1}]^{-1}$ and trajectories of CMA ($\mu = 0.0016$), SWA ($\lambda = 0.995$), and ACMA ($\mu = 0.16$, $m_1 = 0.1592$, $\kappa = 106$) initialized at points $\mathbf{w}(0) = [0\ \ 1]^T$, $\mathbf{w}(0) = [-0.4\ \ 0.05]^T$, and $\mathbf{w}(0) = [0.2\ \ -0.4]^T$.

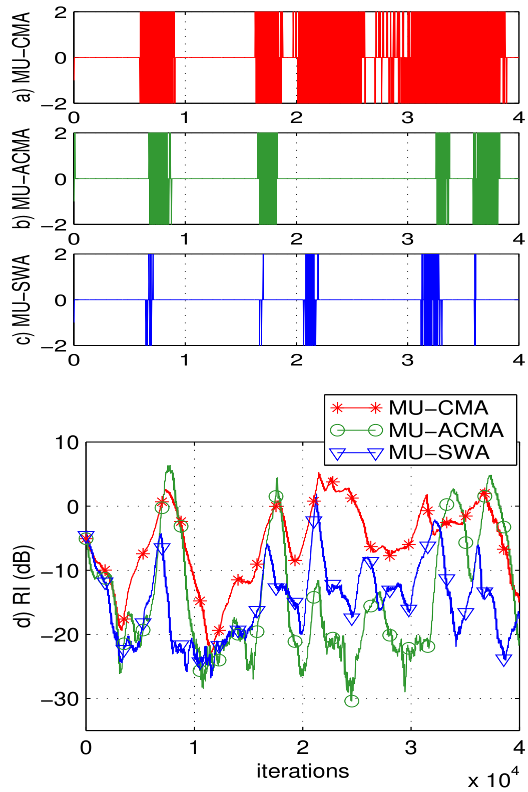

Figure 3: Output errors and RI curves for equalizer-1 using MU-CMA ($\mu = 0.005$, $\xi = 4$), MU-ACMA ($\mu = 0.5$, $m_1 = 0.15$, $\kappa = 100$, $\xi = 4$), and MU-SWA ($\lambda = 0.995$, $\xi = 4$). For 2-PAM, $N = 2$, $L = 3$, $M = 15$, SNR=30 dB, and time-varying channels.

6. CONCLUSIONS

We propose a novel Constant Modulus Algorithm for space-time blind equalization. It is based on the general methodology for the design of blind algorithms [7] and on the accelerating tuner [4].

In [5] an MSE analysis was presented and the stability domain for the AAF algorithm was established. Since J_G is a nonquadratic function, a direct extension of this MSE analysis to MU-ACMA is not a trivial matter. On the other hand, the stability domain of the AAF algorithm was directly extended to MU-ACMA. By means of simulations, MU-ACMA presents adequate behavior in that domain, avoiding instability problems. However, a theoretical analysis is required.

Recently tracking analysis for some blind algorithms was presented considering the energy conservation relation [12]. These results do not include algorithms of the form of ACMA and their extension is an open problem.

In spite of the lack of theoretical results, simulations show the good behavior of MU-ACMA compared to MU-CMA and MU-SWA in terms of tracking capability and computational complexity. Moreover, it was verified that it may avoid soft local minima.

APPENDIX - DERIVATION OF ACMA

Replacing (12) by a nonlinear version for the i^{th} user

$$\varepsilon_i(t) = \varphi(\mathbf{u}^T(t)\mathbf{w}_i(t)) \tag{16}$$

with $\varphi(\cdot)$ satisfying $\frac{\partial\varphi(\mathbf{u}^T\mathbf{w}_i)}{\partial\mathbf{w}_i} = g\left(\mathbf{u}^T\mathbf{w}_i\right)\mathbf{u}$, where $\frac{\partial}{\partial\mathbf{w}_i}$ stands for derivation with respect to the complex vector $\mathbf{w}_i$, and $g(\cdot)$ is a scalar function. Thus, a linear approximation of $\varphi(\cdot)$ may be written as

$$\varphi(\mathbf{u}^T\mathbf{w}_i) \approx \varphi(\mathbf{u}^T\mathbf{w}_i^{\mathrm{o}}) + g(\mathbf{u}^T\mathbf{w}_i^{\mathrm{o}})\mathbf{u}^T(\mathbf{w}_i - \mathbf{w}_i^{\mathrm{o}}). \tag{17}$$

Applying Euler's reverse rule to expressions (10), (11), and (16) we obtain

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mu\mathbf{q}_i(n) \tag{18}$$

$$\begin{aligned}\mathbf{q}_i(n) = \mathbf{q}_i(n-1) - \mu\mathbf{M_1}\Big\{&\mathbf{u}^*(n)\varepsilon_i(n) + \\ &+2\big(\mathbf{M_2} + \mathbf{u}^*(n)\mathbf{u}^T(n)\mathbf{M_1}\mathbf{M_3}\big)\mathbf{q}_i(n)\Big\}\end{aligned} \tag{19}$$

$$\varepsilon_i(n) = \varphi\big(\mathbf{u}^T(n)\mathbf{w}_i(n)\big) \tag{20}$$

where μ is the integration step. These equations do not make the update of $\mathbf{w}_i(n)$ and $\mathbf{q}_i(n)$ possible. To overcome this obstacle we introduce an *a priori* error

$$e_i(n) = \varphi\big(\mathbf{u}^T(n)\mathbf{w}_i(n-1)\big) = \varphi\big(y_i(n)\big). \tag{21}$$

The functions $\varphi(\cdot)$ and $g(\cdot)$ can be chosen based on the "instantaneous version" of the Godard cost functions [10]:

$$\Psi_i(n) = \frac{1}{2(\beta-1)}\left(|y_i(n)|^2 - R_2^a\right)^2. \tag{22}$$

Calculating the gradient vector of this function with respect to $\mathbf{w}_i$ results

$$\nabla_i\Psi(n) = (|y_i(n)|^2 - R_2^a)y_i(n)\mathbf{u}^*(n) = \varphi\left(y_i(n)\right)\mathbf{u}^*(n) \text{ and} \tag{23}$$

$$\frac{\partial\varphi\left(y_i(n)\right)}{\partial\mathbf{w}_i} = \left(\beta|y_i(n)|^2 - R_2^a\right)\mathbf{u}(n) = g(y_i(n))\mathbf{u}(n). \tag{24}$$

With these assumptions, (21) can be rewritten as

$$e_i(n) = (|y_i(n)|^2 - R_2^a)y_i(n). \tag{25}$$

It is relevant to note from (18) that

$$\mathbf{u}^T(n)\mathbf{w}_i(n) = \mathbf{u}^T(n)\mathbf{w}_i(n-1) + \mu\mathbf{u}^T(n)\mathbf{q}_i(n). \tag{26}$$

By means of (26) and (17) the *a posteriori* error $\varepsilon_i(n)$ can be computed from the *a priori* error $e_i(n)$ as follows:

$$\begin{aligned}\varepsilon_i(n) &= \varphi\big(\mathbf{u}^T(n)\mathbf{w}_i(n-1) + \mu\mathbf{u}^T(n)\mathbf{q}_i(n)\big)\\ &\approx \varphi(y_i(n)) + \mu g\left(y_i(n)\right)\mathbf{u}^T(n)\mathbf{q}_i(n)\\ &= e_i(n) + \mu(\beta|y_i(n)|^2 - R_2^a)\mathbf{u}^T(n)\mathbf{q}_i(n).\end{aligned} \tag{27}$$

Using (27) and (19) we obtain an update expression:

$$\begin{aligned}\mathbf{q}_i(n) &= \mathbf{G}_i^{-1}\left(\mathbf{q}_i(n-1) - \mu\mathbf{M_1}\mathbf{u}^*(n)e_i(n)\right)\\ \mathbf{G}_i &= \mathbf{I} + \mu^2(\beta|y_i(n)|^2 - R_2^a)\mathbf{M_1}\mathbf{u}^*(n)\mathbf{u}^T(n) + \\ &\quad + 2\mu\mathbf{M_1}\big(\mathbf{M_2} + \mathbf{u}^*(n)\mathbf{u}^T(n)\mathbf{M_1}\mathbf{M_3}\big).\end{aligned} \tag{28}$$

It can be shown that the inverse of matrix $\mathbf{G}_i$ is given by

$$\mathbf{G}_i^{-1} = \mathbf{A}\left\{\mathbf{I} - \frac{\mathbf{M_1}\mathbf{u}^*(n)\mathbf{u}^T(n)\mathbf{B}_i(n)\mathbf{M_1}^{-1}}{1 + \mathbf{u}^T(n)\mathbf{B}_i(n)\mathbf{u}^*(n)}\right\} \quad \text{with}$$

$$\mathbf{A} = \left(\mathbf{I} + 2\mu\mathbf{M_1}\mathbf{M_2}\right)^{-1} \quad \text{and}$$

$$\mathbf{B}_i(n) = \mu\left\{\mu\left(\beta|y_i(n)|^2 - R_2^a\right)\mathbf{I} + 2\mathbf{M_1}\mathbf{M_3}\right\}\mathbf{A}\mathbf{M_1}.$$

Replacing this result into (28), making $\mathbf{M}_k = m_k\mathbf{I}$, $k = 1, 2, 3$, with m_k being positive constants, we obtain the version of ACMA shown in Table 1.

ACKNOWLEDGMENTS

The authors thank Dr. Luiz A. Baccalá for his suggestions.

REFERENCES

[1] C. B. Papadias and A. J. Paulraj, "A constant modulus algorithm for multiuser signal separation in presence of delay spread using antenna arrays," *IEEE Signal Process. Letters*, vol. 4, pp. 178-181, June 1997.

[2] M. T. M. Silva, M. D. Miranda, and A. N. Licciardi Jr., "A robust algorithm for blind space-time equalization," *Proc. ICASSP*, Montreal, May 2004.

[3] Y. Luo and J. A. Chambers, "Quasi-Newton cross-correlation and constant modulus adaptive algorithm for space-time blind equalization," *Proc. IMA Math. in Signal Process. Conf. Digest*, Warwick University, Dec. 2000.

[4] M. Gerken, F. M. Pait, and P. E. Jojoa Gomez, "An adaptive filtering algorithm with parameter acceleration," *Proc. ICASSP*, vol. I, pp. 17-20, Istambul, Apr. 2000.

[5] P. E. Jojoa Gomez, M. Gerken, and F. M. Pait, "The accelerating adaptive filtering algorithm," *Proc. IFAC ALCOSP*, pp. 331-335, Cernobio-Como, Aug. 2001.

[6] F. M. Pait, "A tuner that accelerates parameters," *Systems & Control Letters*, vol. 35, pp. 65-68, 1998.

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

[8] K. L. Yeung and S. F. Yau, "A cumulant-based super-exponential algorithm for blind deconvolution of multi-input multi-output systems," *Signal Process.*, vol. 67, pp. 141-162, June 1998.

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

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

[11] T. Shimamura and C. F. N. Cowan, "Equalisation of time variant multipath channels using amplitude banded techniques," *Proc. ICASSP*, pp. 2497-2500, Munich, Apr. 1997.

[12] A. H. Sayed, *Fundamentals of Adaptive Filtering*, 1. ed., John Wiley & Sons, New Jersey, 2003.