ADAPTIVE MINIMUM VARIANCE METHODS FOR DIRECT BLIND MULTICHANNEL EQUALIZATION[1]

Zhengyuan Xu *Michail K. Tsatsanis*

Electrical & Computer Engr. Dept.
Stevens Institute of Technology
Castle Point on Hudson
Hoboken, NJ 07030

e-mail: `mtsatsan@stevens-tech.edu`

ABSTRACT

Constrained adaptive optimization techniques are employed in this paper to design direct blind equalizers. The method is based on minimizing the equalizer's output variance subject to appropriate constraints. The constraints are chosen to guarantee no desired signal cancellation and are also jointly and recursively optimized to improve performance. Our method provides adaptive solutions which directly optimize the equalizer's parameters, while its performance compares favorably to that of the linear prediction based approaches. Global convergence is established and comparisons with other blind and trained methods are presented.

1. INTRODUCTION

Direct equalizer design has received considerable attention recently, because it avoids the explicit step of estimating the channel parameters. Adaptive direct methods have been derived based on multichannel linear prediction [1], yielding simple recursive algorithms. However, those methods are naturally suited for the noiseless case and possess no optimality in the presence of noise.

From a different viewpoint, batch constrained optimization methods were developed in [7], which minimize the receiver's output variance subject to appropriate constraints. Such methods originated in array processing (Minimum Variance Distortionless Response and Capon beamformers), and have been proposed for multiuser communication problems in the context of CDMA systems [5], [8]. It was shown in [7], that by jointly optimizing the constraint parameters, a blind method may be derived whose performance is close to that of the trained MMSE equalizer at high SNR. It is worthwhile therefore, to explore adaptive implementations of such techniques, in order to reduce their computational complexity.

In this paper we derive stochastic gradient and RLS based methods for adaptive multichannel equalization based on recursive constrained optimization techniques which originated in [2]. In our case, the set of constraints depends on certain parameters which also have to be optimized. This further complicates the optimization process as well as its convergence analysis. However, global convergence for the proposed algorithms is established. Comparisons with other equalizers such as linear prediction method [3] and the MMSE receiver are also presented.

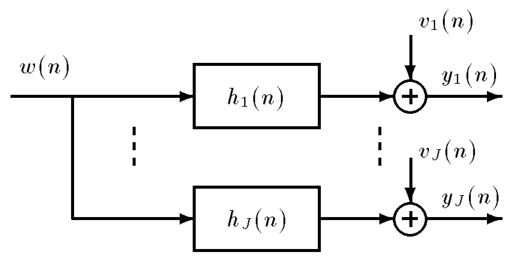

Figure 1. Multichannel model with J antennas

2. MULTICHANNEL MODEL

Consider a communication system with linear modulation and let the user transmit symbols $w(m)$ with period T_s through a multipath FIR channel. Multiple channels may result from employing an antenna array or from oversampling the output signal of the receiver. Assume J antennas are used to get J diversity channels. It can be shown (c.f. [7]) that after sampling at $t = nT_s$, the received discrete-time signal $\mathbf{y}(n) = [y_1(n), \ldots, y_J(n)]^T$ is (see Fig. 1)

$$\mathbf{y}(n) = \sum_{m=-\infty}^{\infty} w(m)\mathbf{h}(n-m) + \mathbf{v}(n) \quad , \tag{1}$$

where $\mathbf{h}(n) = [h_1(n), \ldots, h_J(n)]^T$, $\mathbf{v}(n) = [v_1(n), \ldots, v_J(n)]^T$. Following common practice in communications, we assume that the channels $h_j(k)$ are FIR of order q. Then, if we consider a collection of M successive data vectors $\mathbf{y}_M(n) = [\mathbf{y}^T(n), \ldots, \mathbf{y}^T(n-M+1)]^T$, equation (1) yields

$$\mathbf{y}_M(n) = \mathcal{T}(\mathbf{h})\mathbf{w}_M(n) + \mathbf{v}_M(n) \quad , \tag{2}$$

where

$$\mathcal{T}(\mathbf{h}) = \begin{bmatrix} \mathbf{h}(0) & \ldots & \mathbf{h}(q) & \ldots & \ldots & \mathbf{0} \\ \vdots & \ddots & & \ddots & & \vdots \\ \vdots & & \ddots & & \ddots & \vdots \\ \mathbf{0} & \ldots & \ldots & \mathbf{h}(0) & \ldots & \mathbf{h}(q) \end{bmatrix} \tag{3}$$

is a $(JM) \times (M+q)$ block Toeplitz matrix, and $\mathbf{w}_M(n) = [w(n), \ldots, w(n-M-q+1)]^T$, $\mathbf{v}_M(n) = [\mathbf{v}^T(n), \ldots, \mathbf{v}^T(n-$

[1] This work was supported by the National Science Fundation Grant NSF-NCR 9706658.

$M+1)]^T$. Equations (2), (3) present a compact matrix formulation which will be useful in the development of constrained optimization methods in the sequel.

3. BLIND MINIMUM VARIANCE EQUALIZER

The minimum variance approach to blind equalization is based on the similarity of eq. (2), (3) with corresponding array processing models [7]. Indeed, if $w(n-d)$ is the desired signal (or "source") for a fixed delay d, the other elements of $\mathbf{w}_M(n)$ may be regarded as interfering signals (or "sources"). The signature of the signal $w(n-d)$ is the $d+1$ column of $\mathcal{T}(\mathbf{h})$ and is given by $\mathbf{Ch}$ (c.f. eq. (3)) where

$$\mathbf{C}=\begin{bmatrix}\mathbf{0}_{J(d-q-1)\times J(q+1)}\\ \mathbf{I}_{J(q+1)\times J(q+1)}\\ \mathbf{0}_{J(M-d)\times J(q+1)}\end{bmatrix}, \quad \mathbf{h}=\begin{bmatrix}\mathbf{h}(q)\\ \vdots\\ \mathbf{h}(0)\end{bmatrix}$$

and where $M \geq d+1 \geq q+1$. In order to apply MVDR beamforming ideas in the current setup, we minimize the output variance subject to appropriate constraints such that no desired signal cancellation occurs. In order to force a constant response for the signal of interest, we consider the set of constraints $\mathbf{C}^H\mathbf{f}=\mathbf{u}$, where $\mathbf{u}$ is an arbitrary parameter vector. We therefore arrive at the optimization problem

$$\min_{\mathbf{f}} \mathrm{E}\{\|\hat{w}(n-d)\|^2\} = \min_{\mathbf{f}} \mathbf{f}^H\mathbf{R}_y\mathbf{f} \text{ subject to } \mathbf{C}^H\mathbf{f}=\mathbf{u}, \quad (4)$$

where $\mathbf{R}_y = \mathrm{E}\{\mathbf{y}_M(n)\mathbf{y}_M^H(n)\}$, and $\mathbf{f}=[\mathbf{f}_1^T,\ldots,\mathbf{f}_J^T]^T$ is a multichannel FIR equalizer of length M in each branch $\mathbf{f}_i$, $(i=1,\ldots,J)$. For a given constraint parameter vector $\mathbf{u}$ the minimum variance solution is (eg. [6])

$$\mathbf{f}_{opt}=\mathbf{R}_y^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}\mathbf{u} \quad , \quad (5)$$

while the minimum variance at $\mathbf{f}_{opt}$ becomes

$$\mathrm{J}_{min}=\mathbf{f}_{opt}^H\mathbf{R}_y\mathbf{f}_{opt}=\mathbf{u}^H(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}\mathbf{u} \quad . \quad (6)$$

In order to optimize the constraint parameter vector $\mathbf{u}$, we employ Capon beamforming ideas and maximize the minimum variance in (6)

$$\max_{\mathbf{u}} \mathrm{J}'_{min}=\max_{\mathbf{u}} \frac{\mathbf{u}^H(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}\mathbf{u}}{\mathbf{u}^H\mathbf{u}} \quad . \quad (7)$$

A normalized version of (6) is used in (7) which is insensitive to the length of $\mathbf{u}$. The optimal solution to this problem $\mathbf{u}_{opt}$ is the eigenvector of matrix $(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}$ corresponding to its maximum eigenvalue. Equations (5) and (7) constitute a complete batch equalization algorithm. In the next section, we directly minimize the joint Lagrange cost function with a gradient descent procedure and avoid the explicit computation of $\mathbf{R}_y^{-1}$ to reduce its computational cost.

3.1. Constrained Stochastic Gradient (CSG) Equalizer

Let us write the Lagrangian cost function explicitly as

$$J_1=\mathbf{f}^H\mathbf{R}_y\mathbf{f}+\boldsymbol{\lambda}^H(\mathbf{C}^H\mathbf{f}-\mathbf{u})+(\mathbf{f}^H\mathbf{C}-\mathbf{u}^H)\boldsymbol{\lambda} \quad , \quad (8)$$

where constraints for $\mathbf{f}$ are considered and $\boldsymbol{\lambda}$ is the corresponding Lagrange multiplier. Our goal is to minimize J_1 with respect to $\mathbf{f}$ and maximize it with respect to $\mathbf{u}$, so we obtain two update equations for $\mathbf{f}$ and $\mathbf{u}$ respectively as

$$\mathbf{f}_{n+1}=\mathbf{f}_n-\mu_f\nabla_{\mathbf{f}_n^*}J_1 \quad (9)$$

$$\mathbf{u}_{n+1}=\mathbf{u}_n+\mu_u(\mathbf{I}-\frac{\mathbf{u}_n\mathbf{u}_n^H}{\mathbf{u}_n^H\mathbf{u}_n})\nabla_{\mathbf{u}_n^*}J_1 \quad (10)$$

where μ_f,μ_u are two step sizes. A projection matrix is used in (10), since a change in the length of $\mathbf{u}$ only affects the scaling of $\mathbf{f}$ (see (5)) and has no effect on the performance of the receiver. Thereafter normalization of $\mathbf{u}_{n+1}$ follows

$$\mathbf{u}_{n+1} \leftarrow \frac{\mathbf{u}_{n+1}}{\|\mathbf{u}_{n+1}\|} \quad (11)$$

at each iteration to make the algorithm consistent with the normalized solution to (7) which we seek. In order to obtain our final update equations, we substitute the gradient of (8) in (9),(10). Furthermore, we determine $\boldsymbol{\lambda}_n$ by enforcing the constraint $\mathbf{C}^H\mathbf{f}_{n+1}=\mathbf{u}_n$ at each iteration (see also [2]). Finally we employ instantaneous approximation $\hat{\mathbf{R}}_y(n)=\mathbf{y}_M(n)\mathbf{y}_M^H(n)$ for $\mathbf{R}_y$, and arrive at the recursions

$$\mathbf{f}_{n+1}=(\mathbf{I}-\mathbf{C}\mathbf{C}^H)[\mathbf{f}_n-\mu_f\mathbf{y}_M(n)\mathbf{y}_M^H(n)\mathbf{f}_n]+\mathbf{C}\mathbf{u}_n \quad (12)$$

$$\mathbf{u}_{n+1}=\mathbf{u}_n+\frac{\mu_u}{\mu_f}(\mathbf{I}-\frac{\mathbf{u}_n\mathbf{u}_n^H}{\mathbf{u}_n^H\mathbf{u}_n})\mathbf{C}^H[\mu_f\mathbf{y}_M(n)\mathbf{y}_M^H(n)-\mathbf{I}]\mathbf{f}_n \quad (13)$$

Equations (12), (13) and (11) constitute our CSG based algorithm. Eq. (12) resembles the method of [2]. In the current setup however, the constraints are not fixed, but depend on a parameter vector $\mathbf{u}$ which is also updated. The major difficulty induced that way, is that the cost function depends on the length of $\mathbf{u}$; hence, even at the desired solution, $\nabla_{\mathbf{u}_{opt}^*}J_1 \neq 0$ but is parallel to $\mathbf{u}_{opt}$. This necessitates the projection and/or normization operations of (10), (11). In the sequel, we avoid that problem by explicitly constraining the length of $\mathbf{u}$ in the cost function.

3.1.1. *Alternative Constrained Stochastic Gradient (ACSG) Equalizer*

Let us augment the cost function J_1 in eq. (8) by enforcing the constraint $\|\mathbf{u}\|=1$,

$$J_2=\mathbf{f}^H\mathbf{R}_y\mathbf{f}+\boldsymbol{\lambda}^H(\mathbf{C}^H\mathbf{f}-\mathbf{u})+(\mathbf{f}^H\mathbf{C}-\mathbf{u}^H)\boldsymbol{\lambda}+\rho(\mathbf{u}^H\mathbf{u}-1) \quad . \quad (14)$$

Here, two kinds of lagrange multipliers, the vector $\boldsymbol{\lambda}$ and scalar ρ are involved corresponding to the linear and quadratic constraints. From (14) we may compute the gradients $\nabla_{\mathbf{f}^*}J_2=\mathbf{R}_y\mathbf{f}+\mathbf{C}\boldsymbol{\lambda}$, and $\nabla_{\mathbf{u}^*}J_2=\rho\mathbf{u}-\boldsymbol{\lambda}$, and arrive at two iterative equations for $\mathbf{f}$ and $\mathbf{u}$

$$\mathbf{f}_{n+1}=\mathbf{f}_n-\mu_f(\mathbf{R}_y\mathbf{f}_n+\mathbf{C}\boldsymbol{\lambda}_n) \quad (15)$$

$$\mathbf{u}_{n+1}=\mathbf{u}_n+\mu_u(\rho_n\mathbf{u}_n-\boldsymbol{\lambda}_n) \quad (16)$$

Substituting (15) in $\mathbf{C}^H\mathbf{f}_{n+1} = \mathbf{u}_n$ we solve for $\boldsymbol{\lambda}_n$, and obtain the same update equation for $\mathbf{f}_{n+1}$ as in (12). A different expression however is obtained for $\mathbf{u}_{n+1}$

$$\mathbf{u}_{n+1} = \mu_u \rho_n \mathbf{u}_n + \mathbf{x}_n \tag{17}$$

where $\mathbf{x}_n$ can be expressed as

$$\mathbf{x}_n = \mathbf{u}_n - \frac{\mu_u}{\mu_f}[\mathbf{C}^H(\mathbf{f}_n - \mu_f \mathbf{y}_M(n)\mathbf{y}_M^H(n)\mathbf{f}_n) - \mathbf{u}_n] \tag{18}$$

after using an instantaneous approximation $\hat{\mathbf{R}}_y(n) = \mathbf{y}_M(n)\mathbf{y}_M^H(n)$ for $\mathbf{R}_y$. Imposing the constraint on $\mathbf{u}_{n+1}$ as $\mathbf{u}_{n+1}^H\mathbf{u}_{n+1} = \|\mathbf{u}_{n+1}\|^2 = 1$ and substituting from (17), we can obtain ρ_n by solving a second order equation

$$a_1\rho_n^2 + a_2\rho_n + a_3 = 0 \tag{19}$$

where

$$a_1 = \mu_u^2\|\mathbf{u}_n\|^2, \quad a_2 = \mu_u(\mathbf{u}_n^H\mathbf{x}_n + \mathbf{x}_n^H\mathbf{u}_n), \quad a_3 = \mathbf{x}_n^H\mathbf{x}_n$$

Once ρ_n is obtained, $\mathbf{u}_{n+1}$ can be updated according to (17). This algorithm facilitates our theoretical analysis due to the explicit cost function of (14).

Both methods are LMS based and their convergence depends on the eigenvalue spread of $\mathbf{R}_y$. Hence, they may experience slow convergence rates at high SNR. For this reason we explore RLS based solutions with faster convergence in the next section.

3.2. Blind RLS Equalizer

If we re-consider the cost function (7) from a computational point of view, we can see that the inversion of $\mathbf{R}_y$ constitutes the main bulk of the computational burden. The eigendecomposition step is less demanding since $\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C}$ is a much smaller matrix of size $J(q+1)\times J(q+1)$ compared with $\mathbf{R}_y$ ($JM \times JM$). We may therefore provide a recursive version of (5),(7), by invoking Kalman-RLS methods for updating $\hat{\mathbf{R}}_y^{-1}$ (e.g. [4])

$$\mathbf{k}(n) = \frac{\hat{\mathbf{R}}_y^{-1}(n-1)\mathbf{y}_M(n)}{\nu + \mathbf{y}_M^T(n)\hat{\mathbf{R}}_y^{-1}(n-1)\mathbf{y}_M(n)} \tag{20}$$

$$\hat{\mathbf{R}}_y^{-1}(n) = \frac{1}{\nu}\hat{\mathbf{R}}_y^{-1}(n-1) - \frac{1}{\nu}\mathbf{k}(n)\mathbf{y}_M^T(n)\hat{\mathbf{R}}_y^{-1}(n-1) \tag{21}$$

The forgetting factor ν is chosen close to 1 and $\mathbf{R}_y^{-1}$ is initialized as $\hat{\mathbf{R}}_y^{-1}(0) = \delta^{-1}\mathbf{I}$ (δ is a small positive number). SVD may be performed on the matrix $\mathbf{C}^H\hat{\mathbf{R}}_y^{-1}(n)\mathbf{C}$ at each iteration

$$\mathbf{C}^H\hat{\mathbf{R}}_y^{-1}(n)\mathbf{C} = \mathbf{V}(n)\mathbf{D}(n)\mathbf{V}^H(n) \quad , \tag{22}$$

and $\mathbf{u}$ chosen as the eigenvector corresponding to the minimum eigenvalue. Equations (20) to (22) together with (5) comprise our RLS algorithm.

4. GLOBAL CONVERGENCE

Our joint cost function $\boldsymbol{J}_2$ in (14) is parametrized by both $\mathbf{f}$ and $\mathbf{u}$, so it is not immediately clear that our algorithm enjoys global convergence. In order to show this result in the sequel we identify all the stationary points and check their stability. At each stationary point, we have

$$\nabla_{\mathbf{f}^*}\boldsymbol{J}_2 = \mathbf{R}_y\mathbf{f} + \mathbf{C}\boldsymbol{\lambda} = 0$$

$$\nabla_{\mathbf{u}^*}\boldsymbol{J}_2 = \rho\mathbf{u} - \boldsymbol{\lambda} = 0 \tag{23}$$

Cancelling out $\boldsymbol{\lambda}$ and substituting $\mathbf{f}$ in the constraint $\mathbf{C}^H\mathbf{f} = \mathbf{u}$, we obtain

$$(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}\mathbf{u} = -\rho\mathbf{u} \tag{24}$$

$$\boldsymbol{\lambda} = -(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}\mathbf{u} \tag{25}$$

hence $(-\rho, \mathbf{u})$ is an eigen-pair of $(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}$. Thus, the set of possible equilibrium points includes the desired solution $\mathbf{u}$, which is the eigenvector corresponding to the maximum eigenvalue of $(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}$ (c.f. eq. (7)). Equations (24), (25) indicate however, that the cost function possesses stationary points at all eigenvectors of $(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}$, and not only at the maximum one. We therefore need to investigate the Hessian matrix of $\boldsymbol{J}_2$ at the stationary points to determine their stability. From (23) and (25) we obtain

$$\nabla_{\mathbf{u}^*}^2\boldsymbol{J}_2 = \rho\mathbf{I} + (\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}$$

$$= \mathbf{V}\begin{bmatrix} \rho+\xi_1 & & 0 \\ & \ddots & \\ 0 & & \rho+\xi_{J(q+1)} \end{bmatrix}\mathbf{V}^H \tag{26}$$

where $\mathbf{V}$ contains the eigenvectors of $(\mathbf{C}^H\mathbf{R}_y^{-1}\mathbf{C})^{-1}$ while ξ_i ($i = 1, \ldots, J(q+1)$) represents its eigenvalues. Let us order $\xi_1 \le \xi_2 \le \ldots \le \xi_{J(q+1)}$ without loss of generality. According to (24), at a stationary point we have $\rho = -\xi_i$ for some $i = 1, \ldots, J(q+1)$. We may therefore distinguish the following three cases:

1) If $\rho = -\xi_1 = -\xi_{min}$, then $\rho + \xi_i \ge 0$ $\forall i$ and therefore $\nabla_{\mathbf{u}^*}^2\boldsymbol{J}_2 \ge 0$ indicating a minimum point (c.f. (26)).

2) If $\rho = -\xi_{J(q+1)} = -\xi_{max}$, then $\rho + \xi_i \le 0$ $\forall i$ and therefore $\nabla_{\mathbf{u}^*}^2\boldsymbol{J}_2 \le 0$ indicating a maximum point.

3) If $\rho = -\xi_i$, $1 < i < J(q+1)$, then $\rho + \xi_1 \le 0$ while $\rho + \xi_{J(q+1)} \ge 0$ and hence $\nabla_{\mathbf{u}^*}^2\boldsymbol{J}_2$ is nondefinite, indicating a saddle point.

In conclusion, the algorithm will enjoy a globally convergent maximum point as long as the maximum eigenvalue ξ_{max} has multiplicity equal to one. The latter fact was established in [7] under certain conditions on the equalizer length.

Finally, since for a given $\mathbf{u}$ there exists a unique optimum $\mathbf{f}$ (see eq. (5)), global convergence of $\mathbf{u}$ implies global convergence of $\mathbf{f}$.

5. SIMULATIONS

In our simulations, BPSK modulation and raised cosine pulse shaping filter ($\alpha = 0.25$) were used. The i.i.d. signal taking values $\{-1, +1\}$ was transmitted through a 3-ray multipath channel. Each multipath signal arrived at

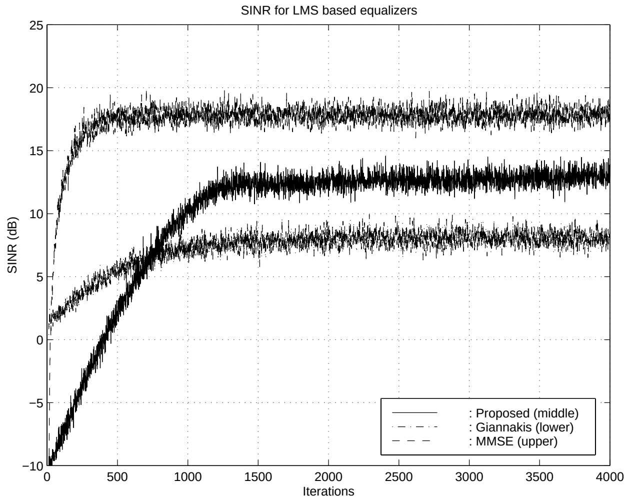

Figure 2. $SINR$ comparison of LMS methods

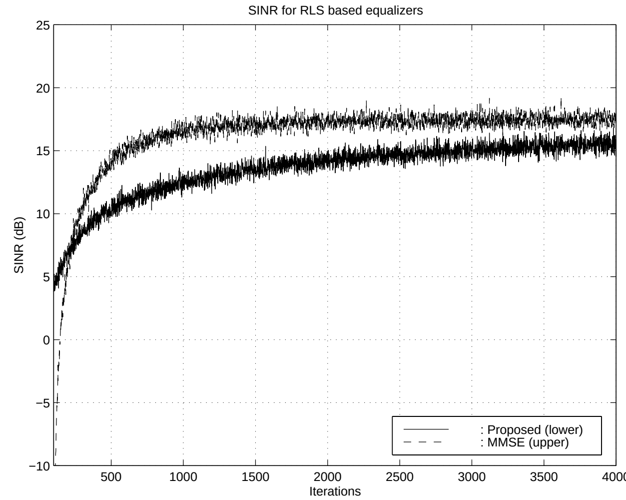

Figure 3. $SINR$ comparison with MMSE of RLS methods

6-antenna array spaced at half wavelength ($\frac{\lambda}{2}$) with a different angle and delay. The signal to white Gaussian noise ratio for each antenna was $10dB$; $d = 5$ and $M = 9$ were chosen which satisfied the identifiability conditions (see [7]). $SINR$ (Signal to Interference and Noise Ratio) versus time was used as our criterion for comparison with other methods.

First we compared our CSG equalizer with the trained MMSE and the linear prediction based adaptive equalizer of [3] in terms of the average output $SINR$ for 100 Monte Carlo runs. As can be seen in Fig. 2 that the proposed method suffers a $4.5dB$ loss when compared with the trained MMSE equalizer but enjoys a $5dB$ gain compared with [3]. Figure 3 compares the proposed RLS method with the trained RLS MMSE equalizer. The lower line shows the $SINR$ for the proposed method, while the upper one for the MMSE solution. We can see that the RLS-version of the method performs closer to the MMSE equalizer than our LMS based algorithm at the expense of increased computational cost. In our last experiment, two different LMS based algorithms of CSG and ACSG were tested under the same conditions and compared in Fig. 4. It is clear from the figure that the $SINR$ of these methods converges to approximately the same level (about $13dB$), but different convergence rates can be observed. The ACSG algorithm converges after 700 iterations according to Fig. (4b) while the other method needs 1100 iterations to converge.

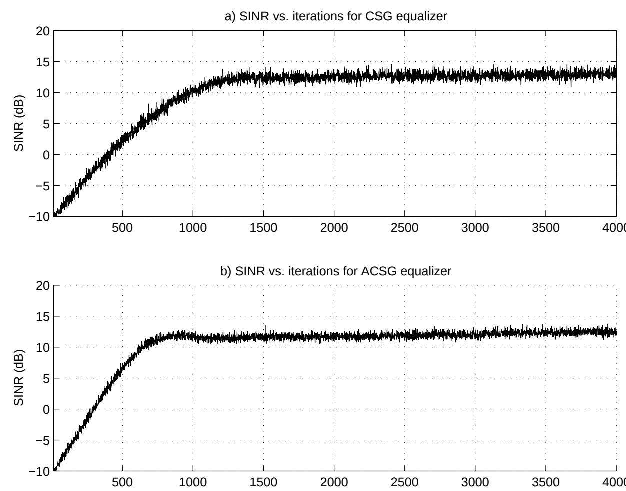

Figure 4. $SINR$ comparison between two proposed LMS based equalizers

REFERENCES

[1] K. Abed Meraim, P. Duhamel, D. Gesbert, P. Loubaton, S. Mayrargue, E. Moulines, and D. Slock, "Prediction Error Methods for Time-Domain Blind Identification of Multichannel FIR Filters", *ICASSP'95,* vol. 3, pp. 1968-1971, Detroit, MI, 1995.

[2] O. L. Frost, "An algorithm for Linearly Constrained Adaptive Array Processing", *Proc. of the IEEE,* vol. 60, no. 8, pp. 926-935, August 1972.

[3] G. B. Giannakis and S. D. Halford, "Blind Fractionally-Spaced Equalization of Noisy FIR Channels: Adaptive and Optimal Solutions", *ICASSP'95,* vol. 3, pp. 1972-1975, Detroit, MI, 1995.

[4] S. Haykin, *Adaptive Filter Theory*, 3rd edition, Prentice Hall, Upper Saddle River, NJ, 1996.

[5] M. Honig, U. Madhow, and S. Verdu, "Blind Adaptive Multiuser Detection", *IEEE Transactions on Information Theory*, vol. 41, no. 4, pp. 944-960, July 1995.

[6] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[7] M. K. Tsatsanis and Z. Xu, "Constrained Optimization Methods for Blind Equalization of Multiple FIR Channels", *31th Asilomar Conf. on Signals, Systems, and Computers,* Pacific Grove, CA, Nov. 2-Nov. 5, 1997.

[8] M. K. Tsatsanis and Z. Xu, "Performance Analysis of Minimum Variance CDMA Receivers", *Proc. 13st Intl. Conf. on Digital Signal Processing (DSP97),* vol. 1, pp. 379-382, Santorini-Hellas, Greece, July 2-4, 1997.