



ELSEVIER

Signal Processing 73 (1999) 125–138

**SIGNAL
PROCESSING**

Adaptive minimum variance methods for direct blind multichannel equalization

Zhengyuan Daniel Xu, Michail K. Tsatsanis*

Electrical and Computer Engineering Department, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030, USA

Received 10 September 1997; received in revised form 6 May 1998

Abstract

In this paper, stochastic gradient and RLS-based methods are presented for designing direct adaptive equalizers. Self-recovering solutions are obtained by 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 adaptively optimized to improve performance. The resulting algorithm may be interpreted as an optimal version of earlier linear prediction-based approaches. It is shown that the algorithm enjoys global convergence. Moreover, the constraint parameters converge to the channel parameters at high SNR. Comparisons with other blind and trained methods are presented. © 1999 Elsevier Science B.V. All rights reserved.

Zusammenfassung

In diesem Beitrag stellen wir "stochastic gradient"- und RLS-basierte Verfahren zur adaptiven direkten Entzerrung vor. Wir präsentieren "self-recovering" Lösungen, die man durch eine Minimierung der Varianz des Entzerrerausgangs unter geeigneten Nebenbedingungen erhält. Die Nebenbedingungen sind derart gewählt, daß keine Nutzsignalanteile entfernt werden. Weiters werden diese Nebenbedingungen gemeinsam adaptiv optimiert, so daß es zu einer Verbesserung der Performance des Entzerrers kommt. Der vorgestellte Algorithmus kann als eine optimale Version von früheren auf linearer Prädiktion basierenden Verfahren interpretiert werden. Wir zeigen, daß der Algorithmus globale Konvergenz aufweist. Bei großem SNR konvergieren die in die Nebenbedingungen eingehenden Parameter gegen die Kanalparameter. Schließlich werden Vergleiche mit anderen blinden Verfahren und Verfahren basierend auf Referenzdaten gezeigt. © 1999 Elsevier Science B.V. All rights reserved.

Résumé

Dans cet article, on présente des méthodes basées sur le gradient stochastique ou sur les moindres carrés récursifs (RLS) pour la construction d'égaliseurs adaptatifs. Des solutions autodidactes sont obtenues en minimisant sous les contraintes appropriées la variance en sortie de l'égaliseur. Les contraintes sont choisies pour éviter l'annulation du signal désiré, et sont conjointement et adaptativement optimisées pour améliorer les performances. L'algorithme résultant peut être interprété comme une version optimale d'approches antérieures de type prédiction linéaire. On montre que l'algorithme converge globalement. En outre, les paramètres de contrainte convergent vers les paramètres du canal

* Corresponding author. Tel.: + 1 201 216 5603; fax: + 1 201 216 8246; e-mail: mtsatsan@stevens-tech.edu

à fort SNR. Des comparaisons avec d'autres méthodes aveugles ou informées sont présentées. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Blind equalization; Adaptive equalization; Multiple FIR channels

1. Introduction

Blind channel identification and equalization have been widely studied from various viewpoints for high speed digital communications in order to combat multipath effects in the absence of training samples [4,5,9,13,14]. Higher-order statistics have been traditionally employed to blindly identify non-minimum phase channels [10]. The pioneering work of Tong et al. [14] however, showed that second-order statistics are sufficient for blind estimation of certain channels, if diversity is achieved either by fractionally sampling the single output of the transmission channel, or by utilizing an antenna array. Estimation of the channel parameters using subspace techniques was demonstrated in [9] while adaptive versions for CDMA systems were investigated in [20]. Recent efforts have focused on direct equalizer design methods, which avoid the explicit step of estimating the channel parameters. Adaptive direct methods have been derived based on multichannel linear prediction [1,4,13], yielding simple recursive algorithms. The methods of [1,13] are naturally suited for the noiseless case and possess no optimality in the presence of noise. Optimal versions in the noisy case were proposed in [17] using constrained optimization principles.

In this paper we focus on solving the direct equalizer design problem by employing constrained adaptive optimization techniques. In particular, we minimize the receiver's output variance, subject to appropriate constraints which guarantee that we avoid the trivial solution. Such methods originated in array processing (minimum variance distortionless response (MVDR) and Capon beamformers [2]), and have been proposed for multiuser communication problems in the context of CDMA systems [7,11,12,15,16]. Honig et al. developed an adaptive method by constraining the desired user's output power to a constant [7]. The method was later extended by adding more constraints [11,12].

The sensitivity of these methods to signal mismatch and multipath effects was addressed in [15], while optimal solutions in the presence of multipath were proposed in [16].

Constrained optimization methods are also applicable to multichannel equalization setups with improved performance over linear prediction methods [17]. In [17], the analysis of batch constrained optimization methods indicated a performance which is close to that of the trained MMSE receiver. 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 [3]. 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.

The main contribution of this paper is that of establishing global convergence for the proposed algorithms. Comparisons with other techniques and the MMSE receiver are also performed.

The rest of the paper is organized as follows. In Section 2 we introduce the multichannel system model. Different constrained adaptive equalizers are proposed in Section 3, whereas in Section 4, the convergence properties of our method are studied. Some simulation results are presented in Section 5 together with comparisons with the trained MMSE and the linear prediction-based equalizer of [4]. Finally Section 6 contains our conclusions.

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.

At the receiver end multiple channels may result from receiving multiple outputs of an antenna array or by oversampling the signal output of a transmission channel. Assume J antennas are used to get J diversity channels, then the received signal at the i th antenna is¹

$$y_{c,i}(t) = \sum_{l=1}^L s_i(\theta_l) a_l \sum_{m=-\infty}^{\infty} w(m) g_c(t - \tau_l - mT_s) + v_{c,i}(t), \quad (1)$$

where $s_i(\theta_l)$ is the gain of the i th antenna for direction θ_l , a_l , τ_l ($l = 1, \dots, L$) are the complex gain and propagation delay respectively of the l th path among a total of L paths, $g_c(t)$ is the impulse response which includes both the transmitter and receiver filters, and $v_{c,i}(t)$ is Gaussian noise added to the i th antenna. If signals from all antennas are collected in a vector $\mathbf{y}_c(t) = [y_{c,1}(t), \dots, y_{c,J}(t)]^T$, then we get a vector form for Eq. (1),

$$\mathbf{y}_c(t) = \sum_{m=-\infty}^{\infty} w(m) \mathbf{h}_c(t - mT_s) + \mathbf{v}_c(t), \quad (2)$$

where $\mathbf{h}_c(t) = \sum_{l=1}^L s(\theta_l) g_c(t - \tau_l)$, $\mathbf{s}(\theta_l) = [a_l s_1(\theta_l), \dots, a_l s_J(\theta_l)]^T$, $\mathbf{v}_c(t) = [v_{c,1}(t), \dots, v_{c,J}(t)]^T$.

After sampling at $t = nT_s$, the received discrete-time signal $\mathbf{y}(n) = \mathbf{y}_c(t)|_{t=nT_s}$ is (see Fig. 1)

$$\mathbf{y}(n) = \sum_{m=-\infty}^{\infty} w(m) \mathbf{h}(n - m) + \mathbf{v}(n), \quad (3)$$

where $\mathbf{h}(n) = \mathbf{h}_c(t)|_{t=nT_s}$, $\mathbf{v}(n) = \mathbf{v}_c(t)|_{t=nT_s}$. Following common practice in communications, we assume that the channels $\mathbf{h}(k)$ are FIR of order q . Then, if we consider a collection of M successive data vectors

$$\mathbf{y}_M(n) = [\mathbf{y}^T(n), \dots, \mathbf{y}^T(n - M + 1)]^T, \quad (4)$$

Eq. (3) yields

$$\mathbf{y}_M(n) = \mathcal{T}(\mathbf{h}) \mathbf{w}_M(n) + \mathbf{v}_M(n), \quad (5)$$

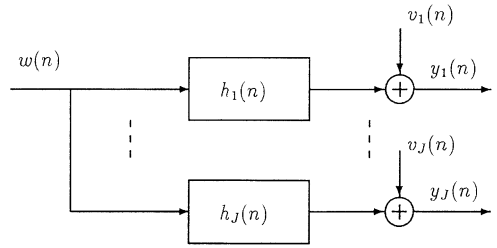


Fig. 1. Multichannel model with J antennas.

where

$$\mathcal{T}(\mathbf{h}) = \begin{bmatrix} \mathbf{h}(0) & \dots & \mathbf{h}(q) & \dots & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{h}(0) & \dots & \mathbf{h}(q) \end{bmatrix} \quad (6)$$

is a $(JM) \times (M + q)$ block Toeplitz matrix, and $\mathbf{w}_M(n) = [w(n), \dots, w(n - M - q + 1)]^T$, $\mathbf{v}_M(n) = [\mathbf{v}^T(n), \dots, \mathbf{v}^T(n - M + 1)]^T$. Eqs. (5) and (6) present a compact matrix formulation which will be useful in the development of constrained optimization methods in the sequel.

3. Blind minimum variance equalizer

In this paper we focus on the design of linear FIR equalizers. Consider the equalizer $\mathbf{f} = [\mathbf{f}_1^T, \dots, \mathbf{f}_J^T]^T$ shown in Fig. 2, where \mathbf{f}_i is a filter of length M ($i = 1, \dots, J$). Then \mathbf{f} is a $(JM \times 1)$ vector. We wish to estimate the transmitted signal $w(n - d)$ with a possible delay of d samples at time n and design a corresponding blind equalizer \mathbf{f} based on our received data $\mathbf{y}_M(n)$, such that the output

$$\hat{w}(n - d) = \mathbf{f}^H \mathbf{y}_M(n) \quad (7)$$

is close to the input symbol $w(n - d)$ in some sense. In the absence of training samples, we explore constrained optimization techniques to derive optimal blind equalizers in what follows. Such techniques have been widely used in beamforming and array signal processing [2,8].

Motivated by the structure of \mathcal{T} , we rewrite Eq. (6) here as

$$\mathcal{T}(\mathbf{h}) = [\mathbf{h}_1, \dots, \mathbf{h}_{d+1}, \dots, \mathbf{h}_{M+q}], \quad (8)$$

¹ We use subscript c to denote continuous-time signals.

Using Lagrange multipliers, the solution to Eq. (17) can be shown to be a function of \mathbf{u} ,

$$\mathbf{f}_{\text{opt}} = \mathbf{R}_y^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1} \mathbf{u}, \quad (18)$$

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

$$\mathbf{J}_{\text{min}} = \mathbf{f}_{\text{opt}}^H \mathbf{R}_y \mathbf{f}_{\text{opt}} = \mathbf{u}^H (\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1} \mathbf{u}. \quad (19)$$

The constraint vector \mathbf{u} may be determined by following Capon's idea [2], i.e., by maximizing \mathbf{J}_{min} with respect to \mathbf{u} . An intuitive justification of this max/min approach relies on maximizing the signal component at the output after the interference has been suppressed.

Let us therefore consider the maximization of the normalized cost function

$$\max_{\mathbf{u}} \mathbf{J}_{\text{min}} = \max_{\mathbf{u}} \frac{\mathbf{u}^H (\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1} \mathbf{u}}{\mathbf{u}^H \mathbf{u}}, \quad (20)$$

which is insensitive to the length of \mathbf{u} . The cost function in Eq. (20) is a Rayleigh quotient, and \mathbf{u}_{opt} is the eigenvector of matrix $(\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1}$ corresponding to its maximum eigenvalue (or the eigenvector corresponding to the minimum eigenvalue of matrix $(\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})$). Eqs. (18) and (20) constitute a complete batch equalization algorithm. Its performance is studied in [17], where the output SINR was observed to be close to that of the trained MMSE receiver. Unfortunately though, the algorithm requires the inversion of \mathbf{R}_y , increasing its computational cost. In the next section, we directly minimize the Lagrange cost function with a gradient descent procedure and avoid the explicit computation of \mathbf{R}_y^{-1} .

3.1. Constrained stochastic gradient (CSG) equalizer

Let us write the Lagrangian cost function explicitly as

$$\mathbf{J}_1 = \mathbf{f}^H \mathbf{R}_y \mathbf{f} + \lambda^H (\mathbf{C}^H \mathbf{f} - \mathbf{u}) + (\mathbf{f}^H \mathbf{C} - \mathbf{u}^H) \lambda, \quad (21)$$

where constraints for \mathbf{f} are considered and λ is the corresponding Lagrange multiplier. Our goal is to minimize \mathbf{J}_1 with respect to \mathbf{f} and maximize \mathbf{J}_1 with respect to \mathbf{u} , so we get two update equations for

\mathbf{f} and \mathbf{u} respectively as

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \mu_f \nabla_{\mathbf{f}_n} \mathbf{J}_1, \quad (22)$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \mu_u \nabla_{\mathbf{u}_n^*} \mathbf{J}_1, \quad (23)$$

where μ_f, μ_u are two step sizes. Since a change in the length of \mathbf{u} only affects the scaling of \mathbf{f} (see Eqs. (18) and (20)) and has no effect on the performance of the receiver, we may project $\nabla_{\mathbf{u}_n^*} \mathbf{J}_1$ onto the space orthogonal to \mathbf{u}_n ,

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \mu_u \left(\mathbf{I} - \frac{\mathbf{u}_n \mathbf{u}_n^H}{\mathbf{u}_n^H \mathbf{u}_n} \right) \nabla_{\mathbf{u}_n^*} \mathbf{J}_1 \quad (24)$$

and/or normalize \mathbf{u}_{n+1} at each iteration

$$\mathbf{u}_{n+1} \leftarrow \frac{\mathbf{u}_{n+1}}{\|\mathbf{u}_{n+1}\|}. \quad (25)$$

This is consistent with the normalized solution to Eq. (20) which we seek. By computing the gradient of Eq. (21) and substituting into Eqs. (22) and (24) we obtain

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \mu_f (\mathbf{R}_y \mathbf{f}_n + \mathbf{C} \lambda_n), \quad (26)$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n - \mu_u \left(\mathbf{I} - \frac{\mathbf{u}_n \mathbf{u}_n^H}{\mathbf{u}_n^H \mathbf{u}_n} \right) \lambda_n, \quad (27)$$

where λ_n is used instead of λ and is adapted at each iteration. To determine λ_n , we substitute Eq. (26) in the constraint $\mathbf{C}^H \mathbf{f}_{n+1} = \mathbf{u}_n$ and solve for λ_n (see also [3])

$$\lambda_n = \frac{1}{\mu_f} (\mathbf{C}^H \mathbf{f}_n - \mu_f \mathbf{C}^H \mathbf{R}_y \mathbf{f}_n - \mathbf{u}_n), \quad (28)$$

where Eq. (12) is used. Substituting Eq. (28) in Eqs. (26) and (27) and using an instantaneous approximation $\hat{\mathbf{R}}_y(n) = \mathbf{y}_M(n) \mathbf{y}_M^H(n)$ for \mathbf{R}_y at each step we obtain

$$\mathbf{f}_{n+1} = \Pi_C^\perp [\mathbf{f}_n - \mu_f \mathbf{y}_M(n) \mathbf{y}_M^H(n) \mathbf{f}_n] + \mathbf{C} \mathbf{u}_n, \quad (29)$$

$$\begin{aligned} \mathbf{u}_{n+1} = \mathbf{u}_n + \frac{\mu_u}{\mu_f} \left(\mathbf{I} - \frac{\mathbf{u}_n \mathbf{u}_n^H}{\mathbf{u}_n^H \mathbf{u}_n} \right) \\ \times [\mu_f \mathbf{C}^H \mathbf{y}_M(n) \mathbf{y}_M^H(n) \mathbf{f}_n - \mathbf{C}^H \mathbf{f}_n], \end{aligned} \quad (30)$$

where

$$\Pi_C^\perp = \mathbf{I} - \mathbf{C} \mathbf{C}^H. \quad (31)$$

Π_C^\perp can be precomputed to reduce computations. Combining Eqs. (29)–(31) and Eq. (25) we summarize our adaptive algorithm as follows.

Step 1. Initialize \mathbf{f} and \mathbf{u} at $n = 0$ and choose two step sizes μ_f, μ_u properly.

Step 2. Use Eq. (31) to compute Π_C^\perp .

Step 3. For $n = 0, 1, 2, \dots$

3.1. collect the received data in a vector \mathbf{y}_M according to Eq. (4);

3.2. use Eq. (29) to compute \mathbf{f}_{n+1} and (30) to get \mathbf{u}_{n+1} ;

3.3. normalize \mathbf{u}_{n+1} as in Eq. (25).

This algorithm is based on the idea of directly optimizing the Lagrangian cost function and choosing λ_n so that the constraints are satisfied at every iteration. In that respect it is similar to the method of [3]. In the current setup, however, the constraints are not fixed, as in [3], but depend on a parameter vector \mathbf{u} which is also optimized. The major difficulty induced that way, is that the cost function depends on the length of \mathbf{u} ; hence, even at the desired solution, $\nabla_{\mathbf{u}^*} \mathbf{J}_1 \neq 0$ but is parallel to \mathbf{u}_{opt} . This necessitates the projection and/or normalization operations of Eqs. (24) and (25). 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 \mathbf{J}_1 in Eq. (21) by enforcing the constraint $\|\mathbf{u}\| = 1$,

$$\mathbf{J}_2 = \mathbf{f}^H \mathbf{R}_y \mathbf{f} + \lambda^H (\mathbf{C}^H \mathbf{f} - \mathbf{u}) + (\mathbf{f}^H \mathbf{C} - \mathbf{u}^H) \lambda + \rho (\mathbf{u}^H \mathbf{u} - 1). \quad (32)$$

Here, two kinds of lagrange multipliers, vector λ and scalar ρ are involved corresponding to the linear and quadratic constraints. From Eq. (32) we may compute the gradients $\nabla_{\mathbf{f}} \mathbf{J}_2 = \mathbf{R}_y \mathbf{f} + \mathbf{C} \lambda$, and $\nabla_{\mathbf{u}} \mathbf{J}_2 = \rho \mathbf{u} - \lambda$, and arrive at two iterative equations for \mathbf{f} and \mathbf{u} ,

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \mu_f (\mathbf{R}_y \mathbf{f}_n + \mathbf{C} \lambda_n), \quad (33)$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \mu_u (\rho_n \mathbf{u}_n - \lambda_n). \quad (34)$$

Substituting Eq. (33) in $\mathbf{C}^H \mathbf{f}_{n+1} = \mathbf{u}_n$ we solve for λ_n similarly to Eq. (28); therefore, the update equation for \mathbf{f}_{n+1} does not change (c.f. Eq. (29)). But ρ_n is needed in Eq. (34) to update \mathbf{u}_n , which in turn is

needed in the update of λ_n and \mathbf{f}_n . If we substitute Eq. (28) in Eq. (34), we arrive at

$$\mathbf{u}_{n+1} = (1 + \mu_u \rho_n) \mathbf{u}_n + \frac{\mu_u}{\mu_f} \mathbf{u}_n - \frac{\mu_u}{\mu_f} \mathbf{x}_n, \quad (35)$$

where

$$\mathbf{x}_n = \mathbf{C}^H (\mathbf{f}_n - \mu_f \mathbf{R}_y \mathbf{f}_n). \quad (36)$$

Imposing constraints on \mathbf{u}_{n+1} as $\mathbf{u}_{n+1}^H \mathbf{u}_{n+1} = \|\mathbf{u}_{n+1}\|^2 = 1$ and substituting from Eq. (35), we can obtain ρ_n by solving a second order equation

$$a_1 \rho_n^2 + a_2 \rho_n + a_3 = 0, \quad (37)$$

where

$$a_1 = \mu_u^2 \|\mathbf{u}_n\|^2,$$

$$a_2 = 2\mu_u \left(1 + \frac{\mu_u}{\mu_f}\right) \|\mathbf{u}_n\|^2 - 2\frac{\mu_u^2}{\mu_f} \text{Re}(\mathbf{u}_n^H \mathbf{x}_n),$$

$$a_3 = \left(1 + \frac{\mu_u}{\mu_f}\right)^2 \|\mathbf{u}_n\|^2 - 2\frac{\mu_u}{\mu_f} \left(1 + \frac{\mu_u}{\mu_f}\right) \text{Re}(\mathbf{u}_n^H \mathbf{x}_n) + \left(\frac{\mu_u}{\mu_f}\right)^2 \|\mathbf{x}_n\|^2 - 1.$$

Here, “Re” stands for the real part of a complex number. There may be two real solutions as long as the discriminant of this equation is nonnegative, that is,

$$\Delta = a_2^2 - 4a_1 a_3 = 4 \frac{\mu_u^4}{\mu_f^2} \left\{ \frac{\mu_f^2}{\mu_u^2} \|\mathbf{u}_n\|^2 + [\text{Re}(\mathbf{u}_n^H \mathbf{x}_n)]^2 - \|\mathbf{u}_n\|^2 \|\mathbf{x}_n\|^2 \right\} \geq 0.$$

It is well-known that $\Delta < 0$ corresponds to the case of incompatible linear and quadratic constraints [19]. In our case however, the optimal \mathbf{f} is a function of \mathbf{u} and corresponds to compatible constraints $\forall \mathbf{u}$. If a stochastic gradient is used (i.e., \mathbf{R}_y is replaced by $\mathbf{y}_M(n) \mathbf{y}_M^H(n)$ in Eq. (36)), then no update for \mathbf{u} should be performed in the occasional situation when $\Delta < 0$. Otherwise, the smaller of the two solutions for ρ should be selected.

$$\rho_n = \frac{1}{2a_1} (-a_2 - \sqrt{\Delta}). \quad (38)$$

Once ρ_n is obtained, \mathbf{u}_{n+1} can be updated according to Eq. (35).

This algorithm has small differences from the one developed in the last section, and has been observed to offer similar performance. It facilitates our theoretical analysis however, due to the explicit cost function of Eq. (32).

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 reconsider the cost function (20) 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 (18), (20) by invoking Kalman-RLS methods for updating $\hat{\mathbf{R}}_y^{-1}$ (e.g. [6])

$$\mathbf{k}(n) = \frac{\hat{\mathbf{R}}_y^{-1}(n-1) \mathbf{y}_M(n)}{\nu + \mathbf{y}_M^T(n) \hat{\mathbf{R}}_y^{-1}(n-1) \mathbf{y}_M(n)} \quad (39)$$

$$\hat{\mathbf{R}}_y^{-1}(n) = \frac{1}{\nu} \hat{\mathbf{R}}_y^{-1}(n-1) - \frac{1}{\nu} \mathbf{k}(n) \mathbf{y}_M^T(n) \hat{\mathbf{R}}_y^{-1}(n-1). \quad (40)$$

The forgetting factor ν is chosen close to 1 and $\hat{\mathbf{R}}_y^{-1}$ 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

$$\mathbf{C}^H \hat{\mathbf{R}}_y^{-1}(n) \mathbf{C} = \mathbf{V}(n) \mathbf{D}(n) \mathbf{V}^H(n), \quad (41)$$

and \mathbf{u} chosen as the eigenvector corresponding to the minimum eigenvalue. Alternatively, the Cholesky factors of $\mathbf{C}^H \hat{\mathbf{R}}_y^{-1}(n) \mathbf{C}$ may be directly updated (see [8] (p. 408) for detail). Eqs. (39)–(41) together with Eq. (18) comprise our RLS algorithm.

3.3. Blind GSC equalizer

Our last variation of the adaptive algorithm transforms the problem to an equivalent uncon-

strained one. Corresponding methods in array processing are called ‘Generalized sidelobe Cancellers’ (GSC).

The constraint matrix \mathbf{C} consists of orthonormal columns, hence it qualifies as a basis for the constraint subspace. Let \mathbf{C}_n be a collection of orthogonal basis vectors which span the orthogonal complement of the constraint subspace. Then the columns of \mathbf{C} and \mathbf{C}_n span the entire space, so we can express \mathbf{f} as $\mathbf{f} = \mathbf{C}\mathbf{s} - \mathbf{C}_n\mathbf{v}$. Using \mathbf{f} in the constraint $\mathbf{C}^H \mathbf{f} = \mathbf{u}$ we obtain $\mathbf{s} = \mathbf{u}$, and

$$\mathbf{f} = \mathbf{C}\mathbf{u} - \mathbf{C}_n\mathbf{v}. \quad (42)$$

After some manipulations our cost function can be written as

$$\begin{aligned} \mathbf{J}_3 = \mathbf{f}^H \mathbf{R}_y \mathbf{f} = & \mathbf{u}^H \mathbf{C}^H \mathbf{R}_y \mathbf{C} \mathbf{u} + \mathbf{v}^H \mathbf{C}_n^H \mathbf{R}_y \mathbf{C}_n \mathbf{v} \\ & - \mathbf{u}^H \mathbf{C}^H \mathbf{R}_y \mathbf{C}_n \mathbf{v} - \mathbf{v}^H \mathbf{C}_n^H \mathbf{R}_y \mathbf{C} \mathbf{u}. \end{aligned} \quad (43)$$

We want to minimize \mathbf{J}_3 with respect to \mathbf{v} and maximize \mathbf{J}_3 with respect to \mathbf{u} to obtain our optimal equalizer. Therefore, we can construct two update equations for \mathbf{v} and \mathbf{u} respectively as

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \mu_v \nabla_{\mathbf{v}} \mathbf{J}_3, \quad (44)$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \mu_u \nabla_{\mathbf{u}} \mathbf{J}_3. \quad (45)$$

If we compute the gradients from Eq. (43), then the two update equations become

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \mu_v \mathbf{C}_n^H \mathbf{R}_y \mathbf{C}_n \mathbf{v}_n + \mu_v \mathbf{C}_n^H \mathbf{R}_y \mathbf{C} \mathbf{u}_n, \quad (46)$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \mu_u \mathbf{C}^H \mathbf{R}_y \mathbf{C} \mathbf{u}_n - \mu_u \mathbf{C}^H \mathbf{R}_y \mathbf{C}_n \mathbf{v}_n, \quad (47)$$

\mathbf{u}_n needs to be normalized at each step, therefore we compute

$$\hat{\mathbf{u}}_{n+1} = \frac{\mathbf{u}_{n+1}}{\|\mathbf{u}_{n+1}\|}. \quad (48)$$

Once $\hat{\mathbf{u}}_{n+1}$, \mathbf{v}_{n+1} are obtained at iteration $n+1$, our GSC equalizer can be constructed according to Eq. (42) as

$$\mathbf{f}_{n+1} = \mathbf{C} \hat{\mathbf{u}}_{n+1} - \mathbf{C}_n \mathbf{v}_{n+1}. \quad (49)$$

Eqs. (46)–(49) together with the approximation to \mathbf{R}_y by $\mathbf{y}_M(n) \mathbf{y}_M^H(n)$ at each iteration form our blind adaptive GSC algorithm.

We may further simplify this algorithm by cancelling out \mathbf{v}_{n+1} at each step. Assume \mathbf{u}_n is

normalized, then

$$\mathbf{f}_n = \mathbf{C}\mathbf{u}_n - \mathbf{C}_n\mathbf{v}_n,$$

from which

$$\mathbf{C}_n\mathbf{v}_n = \mathbf{C}\mathbf{u}_n - \mathbf{f}_n \quad (50)$$

is satisfied. Substituting Eq. (50) in Eq. (47) we can express \mathbf{u}_{n+1} in terms of \mathbf{u}_n and \mathbf{f}_n resulting in a simplified recursion for \mathbf{u}_n ,

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \mu_u \mathbf{C}^H \mathbf{R}_y \mathbf{f}_n. \quad (51)$$

Likewise for $\mathbf{C}_n\mathbf{v}_{n+1}$ according to Eq. (46)

$$\mathbf{C}_n\mathbf{v}_{n+1} = \mathbf{C}\mathbf{u}_n - \mathbf{f}_n + \mu_v \mathbf{C}_n \mathbf{C}_n^H \mathbf{R}_y \mathbf{f}_n.$$

Therefore Eq. (49) becomes

$$\begin{aligned} \mathbf{f}_{n+1} = \mathbf{C} & \frac{\mathbf{u}_n + \mu_u \mathbf{C}^H \mathbf{R}_y \mathbf{f}_n}{\|\mathbf{u}_n + \mu_u \mathbf{C}^H \mathbf{R}_y \mathbf{f}_n\|} \\ & - \mathbf{C}\mathbf{u}_n + \mathbf{f}_n - \mu_v \mathbf{C}_n \mathbf{C}_n^H \mathbf{R}_y \mathbf{f}_n. \end{aligned} \quad (52)$$

Using the fact that $\mathbf{C}_n \mathbf{C}_n^H = \mathbf{I} - \mathbf{C}\mathbf{C}^H$ in Eq. (52),

$$\begin{aligned} \mathbf{f}_{n+1} = \mathbf{f}_n - \mu_v \mathbf{R}_y \mathbf{f}_n \\ & + \left(\mu_v + \frac{\mu_u}{\|\mathbf{u}_n + \mu_u \mathbf{C}^H \mathbf{R}_y \mathbf{f}_n\|} \right) \mathbf{C}\mathbf{C}^H \mathbf{R}_y \mathbf{f}_n \\ & + \left(\frac{1}{\|\mathbf{u}_n + \mu_u \mathbf{C}^H \mathbf{R}_y \mathbf{f}_n\|} - 1 \right) \mathbf{C}\mathbf{u}_n. \end{aligned} \quad (53)$$

Eqs. (51)–(53) comprise our simplified GSC algorithm which only involves two vectors $\mathbf{u}_{n+1}, \mathbf{f}_{n+1}$ to be recursively computed.

4. Convergence analysis

The output variance $\mathbf{f}^H \mathbf{R}_y \mathbf{f}$ is a quadratic function of \mathbf{f} , and one might expect that it would present no difficulty in its convergence analysis. Unfortunately, the current cost function (32) is also parametrized by \mathbf{u} and it does not possess a unique stationary point. Our analysis however, reveals that all the stationary points are unstable except for the one corresponding to the desired solution. Hence, the method does enjoy global convergence.

The way we proceed to show this result is to identify all the stationary points and check their stability.

4.1. Global convergence

If we compute the gradients of \mathbf{J}_2 in Eq. (32) with respect to \mathbf{f} and \mathbf{u} and set them equal to zero we obtain

$$\begin{aligned} \nabla_{\mathbf{f}} \mathbf{J}_2 &= \mathbf{R}_y \mathbf{f} + \mathbf{C}\boldsymbol{\lambda} = 0, \\ \nabla_{\mathbf{u}} \mathbf{J}_2 &= \rho \mathbf{u} - \boldsymbol{\lambda} = 0. \end{aligned} \quad (54)$$

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

$$(\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1} \mathbf{u} = -\rho \mathbf{u}, \quad (55)$$

$$\boldsymbol{\lambda} = -(\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1} \mathbf{u}, \quad (56)$$

hence $(-\rho, \mathbf{u})$ is an eigenpair 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. (20)). Eqs. (55) and (56) 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.

Before investigating the stability of those points, let us mention that the output variance can be written as (see Eqs. (19) and (55))

$$\mathbf{u}^H (-\rho \mathbf{u}) = -\rho,$$

which shows that ρ has to be non-positive. That is why the smaller solution for ρ should be chosen in Eq. (38) to maximize J_{\min} at each iteration.

Let us next compute the Hessian matrix of \mathbf{J}_2 at the stationary points. From Eqs. (54) and (56) we obtain

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

which yields

$$\begin{aligned} \nabla_{\mathbf{u}}^2 \mathbf{J}_2 &= \rho \mathbf{I} + (\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1} \\ &= \mathbf{V} \begin{bmatrix} \rho + \xi_1 & & 0 \\ & \ddots & \\ 0 & & \rho + \xi_{J(q+1)} \end{bmatrix} \mathbf{V}^H, \end{aligned} \quad (57)$$

where \mathbf{V} contains the eigenvectors of $(\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1}$ while ξ_i ($i = 1, \dots, J(q+1)$) represents its eigenvalues. Let us order $\xi_1 \leq \xi_2 \leq \dots \leq \xi_{J(q+1)}$ without loss of generality. According to Eq. (55), at a stationary point we have $\rho = -\xi_i$ for some

$i = 1, \dots, J(q + 1)$. We may therefore distinguish the following three cases:

1. If $\rho = -\xi_1 = -\xi_{\min}$, then $\rho + \xi_i \geq 0 \forall i$ and therefore $\nabla_{\mathbf{u}^*}^2 \mathbf{J}_2 \geq 0$ indicating a minimum point (c.f. Eq. (57)).
2. If $\rho = -\xi_{J(q+1)} = -\xi_{\max}$, then $\rho + \xi_i \leq 0 \forall i$ and therefore $\nabla_{\mathbf{u}^*}^2 \mathbf{J}_2 \leq 0$ indicating a maximum point.
3. If $\rho = -\xi_i, 1 < i < J(q + 1)$, then $\rho + \xi_1 \leq 0$ while $\rho + \xi_{J(q+1)} \geq 0$ and hence $\nabla_{\mathbf{u}^*}^2 \mathbf{J}_2$ is non-definite, 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 [17] for high SNR under certain conditions on the equalizer length.

Finally, since for a given \mathbf{u} there exists a unique optimum \mathbf{f} (see Eq. (18)), global convergence of \mathbf{u} implies global convergence of \mathbf{f} .

4.2. Convergence in the mean

In this section, we explore the stability of the proposed recursions and discuss the selection of appropriate step sizes. We focus on the algorithm of Section 3.1.1, while the other stochastic gradient variations are similarly treated.

Let us take the expected value of both sides of Eq. (29),

$$E[\mathbf{f}_{n+1}] = \mathbf{\Pi}_C^\perp \{E[\mathbf{f}_n] - \mu_f \mathbf{R}_y E[\mathbf{f}_n]\} + \mathbf{C}E[\mathbf{u}_n], \quad (58)$$

where we have used the independence assumption (e.g. [6]). The optimal value for \mathbf{f} is

$$\mathbf{f}_{\text{opt}} = \mathbf{R}_y^{-1} \mathbf{C}(\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1} \mathbf{u}_{\text{opt}} = \xi_{\max} \mathbf{R}_y^{-1} \mathbf{C} \mathbf{u}_{\text{opt}} \quad (59)$$

and \mathbf{u}_{opt} is the eigenvector of $(\mathbf{C}^H \mathbf{R}_y^{-1} \mathbf{C})^{-1}$ corresponding to its maximum eigenvalue ξ_{\max} . If we introduce $\Delta \mathbf{f}_n = E[\mathbf{f}_n] - \mathbf{f}_{\text{opt}}$, $\Delta \mathbf{u}_n = E[\mathbf{u}_n] - \mathbf{u}_{\text{opt}}$, and recall Eqs. (31) and (12), then the following can be easily verified:

$$(\mathbf{\Pi}_C^\perp)^2 = \mathbf{\Pi}_C^\perp, \quad \mathbf{\Pi}_C^\perp \mathbf{f}_{\text{opt}} = \mathbf{f}_{\text{opt}} - \mathbf{C} \mathbf{u}_{\text{opt}},$$

$$\mathbf{\Pi}_C^\perp \mathbf{C} E[\mathbf{u}_{n+1}] = 0, \quad \mathbf{\Pi}_C^\perp \mathbf{R}_y \mathbf{f}_{\text{opt}} = 0, \quad \mathbf{\Pi}_C^\perp \mathbf{C} \mathbf{u}_{\text{opt}} = 0.$$

Then, subtracting Eq. (59) from both sides of Eq. (58), the equation for the difference process can

be shown to be

$$\Delta \mathbf{f}_{n+1} = (\mathbf{\Pi}_C^\perp - \mu_f \mathbf{\Pi}_C^\perp \mathbf{R}_y) \Delta \mathbf{f}_n + \mathbf{C} \Delta \mathbf{u}_n. \quad (60)$$

Similarly, taking expected values on both sides of Eq. (35) and invoking the independence assumption we obtain

$$E[\mathbf{u}_{n+1}] = \left(1 + \mu_u \bar{\rho}_n + \frac{\mu_u}{\mu_f}\right) E[\mathbf{u}_n] + \mu_u \mathbf{C}^H \mathbf{R}_y E[\mathbf{f}_n] - \frac{\mu_u}{\mu_f} \mathbf{C}^H E[\mathbf{f}_n], \quad (61)$$

where $\bar{\rho}_n = E[\rho_n]$. Then, subtracting \mathbf{u}_{opt} from both sides we obtain

$$\Delta \mathbf{u}_{n+1} = \left(1 + \mu_u \bar{\rho}_n + \frac{\mu_u}{\mu_f}\right) \Delta \mathbf{u}_n + \left(\mu_u \mathbf{C}^H \mathbf{R}_y - \frac{\mu_u}{\mu_f} \mathbf{C}^H\right) \Delta \mathbf{f}_n + \mu_u (\xi_{\max} + \bar{\rho}_n) \mathbf{u}_{\text{opt}}. \quad (62)$$

By combining Eqs. (60) and (62) we can get a matrix recursion

$$\begin{bmatrix} \Delta \mathbf{f}_{n+1} \\ \Delta \mathbf{u}_{n+1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta \mathbf{f}_n \\ \Delta \mathbf{u}_n \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mu_u (\xi_{\max} + \bar{\rho}_n) \mathbf{u}_{\text{opt}} \end{bmatrix}, \quad (63)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{\Pi}_C^\perp (\mathbf{I} - \mu_f \mathbf{R}_y) & \mathbf{C} \\ (\mu_u / \mu_f) \mathbf{C}^H (\mu_f \mathbf{R}_y - \mathbf{I}) & (1 + \mu_u \bar{\rho}_n + \mu_u / \mu_f) \mathbf{I} \end{bmatrix}.$$

Eq. (63) implies that the stability of the proposed method depends on the stability of matrix \mathbf{A} . Therefore the step sizes μ_u, μ_f should be chosen such that the eigenvalues of $\mathbf{A}^H \mathbf{A}$ have magnitude less than one. Unfortunately, the problem is further involved due to the presence of $\bar{\rho}_n$ in \mathbf{A} ; $\bar{\rho}_n$ is not updated through a linear recursion and the study of its trajectory is intractable. Still, some insight on the stability of the algorithm in the neighborhood of the desired solution can be obtained by letting $\bar{\rho}_n \simeq -\xi_{\max}$.

Eq. (63) also indicates some explicit necessary conditions on the step sizes. The submatrix

$\mathbf{P}_C^\perp(\mathbf{I} - \mu_f \mathbf{R}_y)$ is required to be stable, which implies that $|1 - \mu_f \lambda_i| < 1 \forall i$, where λ_i 's are the eigenvalues of \mathbf{R}_y (see [3]). This may provide a guideline for selecting μ_f . Also, the submatrix $(1 - \mu_u \xi_{\max} + \mu_u/\mu_f)\mathbf{I}$ needs to be stable, therefore $|1 - \mu_u \xi_{\max} + (\mu_u/\mu_f)| < 1$. The later necessary condition can be used to determine μ_u .

5. Simulations

We tested the stochastic gradient as well as the RLS-based methods on simulated data. The simulation results confirm the applicability and performance of the proposed algorithms. We used the output SINR (signal to interference and noise ratio) as a figure of merit and compared the proposed methods with the trained MMSE equalizer and the linear prediction based blind adaptive equalizer in [4]. All our methods were tested on a communication system with BPSK modulation and raised cosine pulse shaping filter ($\alpha = 0.25$). The i.i.d. signal taking values $\{-1, +1\}$ was transmitted through a 3-ray multipath channel. Each multipath

signal arrived with angle and delay -30° , 15° , -4° and 0, $1.1T_s$, $1.6T_s$ (T_s is the symbol period) respectively at 6-antenna array spaced at half wavelength ($\lambda/2$). The signal to white Gaussian noise ratio was 10 dB. In our simulations $d = 5$ and $M = 9$ were chosen which satisfied the identifiability conditions (see [17]).

First we implemented our constrained stochastic gradient (CSG) equalizer and compared its performance with the trained MMSE and the linear prediction based adaptive equalizer of [4]. Fig. 3 depicts the average output SINR (of $K = 100$ Monte Carlo runs) for each method versus time. The average SINR at the i th iteration is defined as (see Eq. (13))

$$\begin{aligned} \text{SINR}_{\text{av}}[i] &= \frac{\sum_{r=1}^K \|\mathbf{f}_r^H[i] \mathbf{C} \mathbf{h}_{i,w_r}(i-d)\|^2}{\sum_{r=1}^K \|\mathbf{f}_r^H[i] \mathbf{y}_{M,r}[i] - \mathbf{f}_r^H[i] \mathbf{C} \mathbf{h}_{i,w_r}(i-d)\|^2} \end{aligned} \quad (64)$$

where K is the number of Monte Carlo runs, and the subscript r indicates that the associated variable depends on the particular run. The MMSE trained

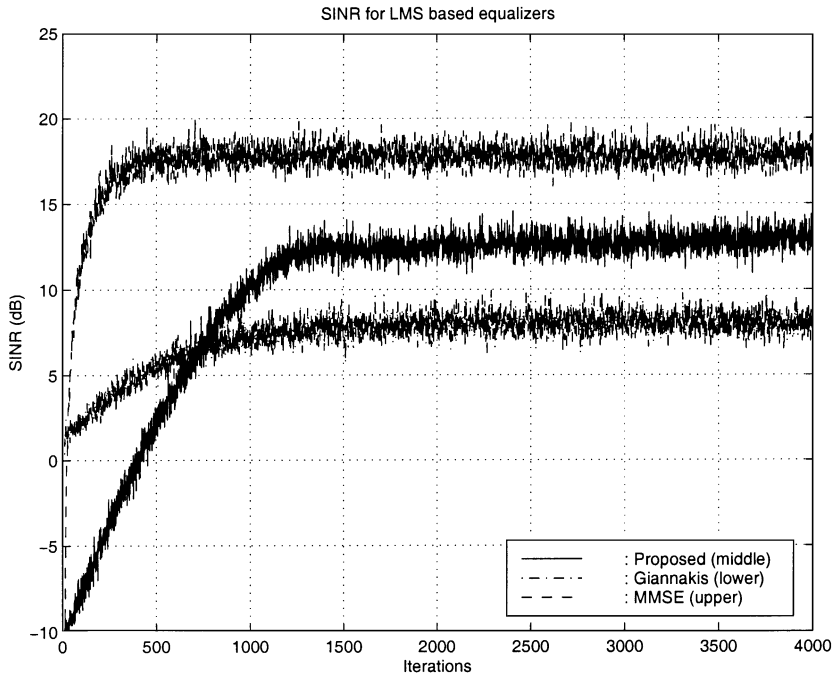


Fig. 3. SINR comparison of LMS methods.

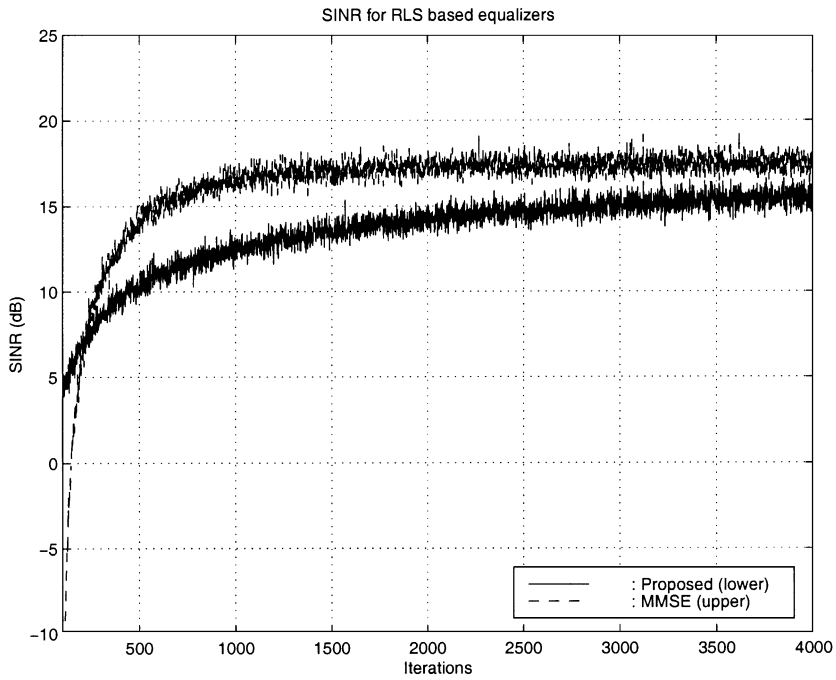


Fig. 4. SINR comparison with MMSE of RLS methods.

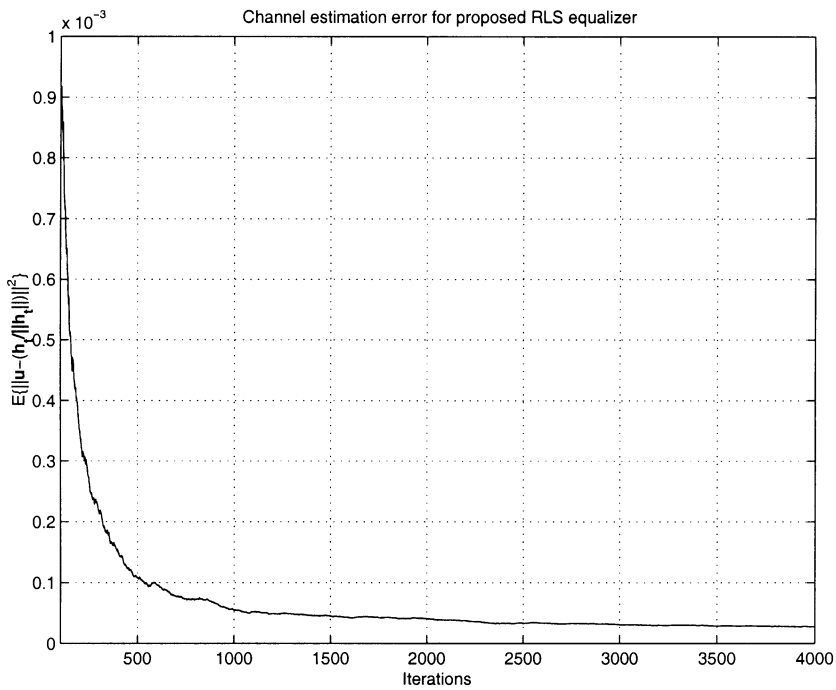


Fig. 5. Channel estimation error for proposed RLS method.

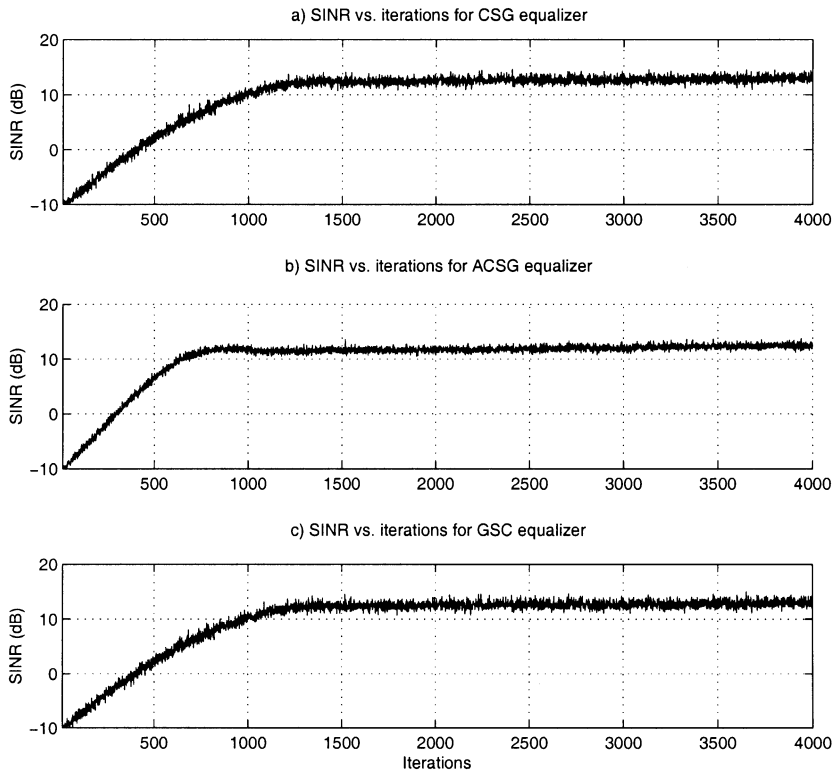


Fig. 6. SINR comparison among three proposed LMS-based equalizers.

equalizer was implemented according to the recursion (see [6])

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \mu_m \mathbf{y}_M(n) [w^*(n-d) - \mathbf{y}_M^H(n) \mathbf{f}_n],$$

and the cyclic LMS equalizer for [4] according to

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \mu_G [\mathbf{y}_M^*(n) \mathbf{y}_M^T(n) \mathbf{f}_n - \sigma_w^2 \mathbf{H}^*(:,1)],$$

where μ_m , μ_G are step sizes, $w(n-d)$ is the desired symbol which is delayed d samples, $\sigma_w^2 = E\{w(n)w^*(n)\}$, $\mathbf{H}^*(:,1)$ is the first column of matrix \mathbf{H} as presented there, $*$ represents complex conjugate and T denotes transpose. The solid line in the middle of three lines represents the result of proposed method as described in Section 3.1, while the upper dashed line and the lower dash-dotted line stand for the performance of the trained MMSE and adaptive equalizer of [4], respectively. If we compute the theoretical SINR using the true data correlation matrix \mathbf{R}_y , we get results for the MMSE and proposed method as 17.5 dB and 17.1 dB,

respectively. Despite the expected similar performance of the two methods in their batch form, it is clear from this figure that there is a 4 dB loss for the proposed method, while the SINR for the adaptive MMSE equalizer is similar to the ideal one. This performance loss is explained by the greater misadjustment of the blind method due to the fact that the blind cost function is orders of magnitude greater than the MSE one, even at the optimum point. Therefore, one should consider switching to a decision feedback algorithm once the eye is open. However, when compared with the method of [4], the proposed method converges to a point which is about 5 dB higher. This fact is not surprising since the linear prediction based methods can be shown to correspond to suboptimal versions of the current method, where the constraint is a priori fixed and not optimized.

Next we investigate the behavior of proposed RLS method. Fig. 4 compares the output SINR of our RLS algorithm presented in Section 3.2 with

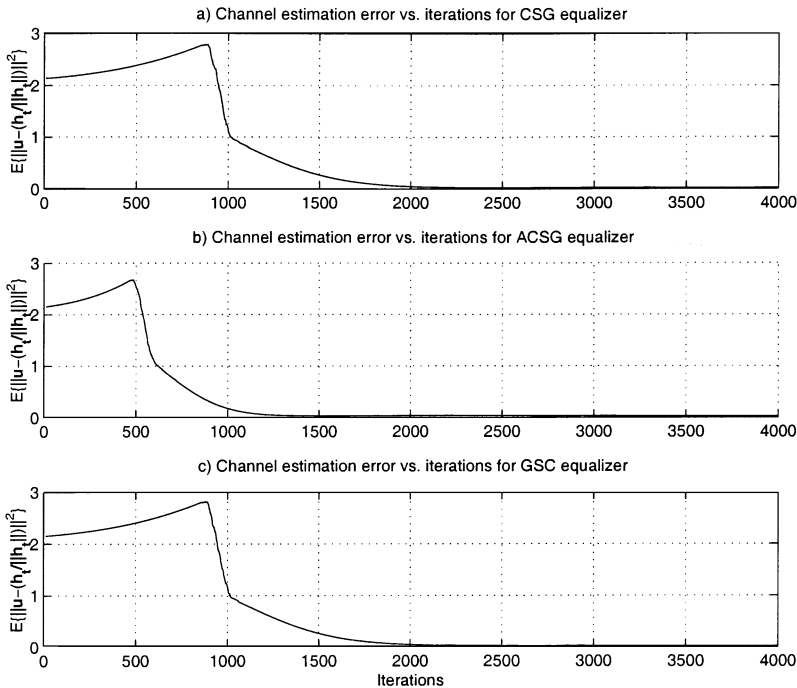


Fig. 7. Channel estimation error for three proposed LMS-based equalizers.

that of the trained MMSE equalizer. The MMSE equalizer is now updated as (see [6])

$$\mathbf{f}_n = \mathbf{f}_{n-1} + \mathbf{k}(n)e^*(n),$$

where $e(n)$ is the error at iteration n ,

$$e(n) = w(n-d) - \mathbf{f}_{n-1}^T \mathbf{y}_M(n),$$

and $\mathbf{k}(n)$ is the Kalman gain from Eq. (39). The lower line shows the results 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. If we compare it with the results from the LMS based CSG algorithm in Fig. 3, it can be seen that it converges to a higher signal level (about 16 dB) with smaller variance. The average channel estimation error $E\{\|\mathbf{u} - (\mathbf{h}_t/\|\mathbf{h}_t\|)\|^2\}$ is shown in Fig. 5, which is very close to 0 (in the order of 10^{-3}). These results are expected since the RLS method exploits all previous data points and does not rely on a stochastic gradient. The RLS algorithm provides better performance at the expense of increased computational cost.

In our last experiment, three different LMS based algorithms of CSG (Section 3.1), ACSG (Section 3.1.1) and GSC (Section 3.3) were tested under the same conditions. We compared them in terms of output SINR in Fig. 6 and in terms of channel estimation error in Fig. 7. Fig. 6(a) shows the SINR corresponding to CSG, while Fig. 6(b) and 6(c) that of ACSG and GSC, respectively. It is clear from the figure that the SINR of these methods converges to approximately the same level (about 13 dB). Similar performances for the channel estimation error $E\{\|\mathbf{u} - (\mathbf{h}_t/\|\mathbf{h}_t\|)\|^2\}$ were also observed in Fig. 7(a), 7(b), 7(c), respectively. But different convergence rates can be seen clearly. The ACSG algorithm converges after 700 iterations according to both Fig. 6(a) and Fig. 7(b) while the other two methods need 1100 iterations to converge.

6. Conclusions

Constrained optimization methods for multi-channel equalization can be adaptively imple-

mented, based on either stochastic gradient or RLS approaches. While global convergence can be guaranteed, blind LMS approaches suffer from a considerable performance loss (compared with trained methods) due to increased misadjustment. Still the proposed method is an optimal version of linear prediction based blind methods. Furthermore, improved performance may be achieved through RLS implementations at the cost of higher complexity.

Acknowledgements

This work was supported in part by the National Science Foundation Grant NSF-NCR 9706658, NSF-CCR 9733048 and the Army Research Office Contract DAAG55-98-1-0224.

References

- [1] K. Abed Meraim, E. Moulines, P. Loubaton, Prediction error method for second-order blind identification, *IEEE Trans. Signal Processing* 45 (3) (March 1997) 694–705.
- [2] J. Capon, High-resolution frequency-wavenumber spectrum analysis, *Proc. IEEE* 57 (8) (August 1969) 2408–2418.
- [3] O.L. Frost, An algorithm for linearly constrained adaptive array processing, *Proc. IEEE* 60 (8) (August 1972) 926–935.
- [4] G.B. Giannakis, S.D. Halford, Blind fractionally-spaced equalization of noisy FIR channels: Adaptive and optimal solutions, *IEEE Trans. Signal Processing* 45 (9) (September 1997) 2277–2292.
- [5] D.N. Godard, Self-recovering equalization and carrier-tracking in two-dimensional data communication Systems, *IEEE Trans. Commun.* 28 (1980) 1867–1875.
- [6] S. Haykin, *Adaptive Filter Theory*, 3rd edition, Prentice Hall, Upper Saddle River, NJ, 1996.
- [7] M. Honig, U. Madhow, S. Verdu, Blind adaptive multiuser detection, *IEEE Trans. Inform. Theory* 41 (4) (July 1995) 944–960.
- [8] D.H. Johnson, D.E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [9] E. Moulines, P. Duhamel, J.-F. Cardoso, S. Mayrargue, Subspace methods for the blind identification of multichannel FIR filters, *IEEE Trans. Signal Processing* 43 (2) (February 1995) 516–525.
- [10] B. Porat, B. Friedlander, Blind equalization of digital communication channels using higher-order statistics, *IEEE Trans. Signal Processing* 39 (1991) 522–526.
- [11] J.B. Schodorf, D.B. Williams, Array processing techniques for multiuser detection, *IEEE Trans. Commun.* 45 (11) (November 1997) 1375–1378.
- [12] J.B. Schodorf, D.B. Williams, A constrained optimization approach to multiuser detection, *IEEE Trans. Signal Processing* 45 (1) (January 1997) 258–262.
- [13] D.T.M. Slock, Blind fractionally-spaced equalization, perfect-reconstruction filter banks and multichannel prediction, in: *Proc. ICASSP 94 Conf.*, Vol. IV, Adelaide, Australia, April 1994, pp. 585–588.
- [14] L. Tong, G. Xu, T. Kailath, Blind identification and equalization based on second-order statistics: A time-domain approach, *IEEE Trans. Inform. Theory* 40 (2) (March 1994) 340–349.
- [15] M.K. Tsatsanis, Inverse filtering criteria for CDMA systems, *IEEE Trans. Signal Processing* (special issue on SP for advanced communications) 45 (1) (January 1997) 102–112.
- [16] M.K. Tsatsanis, Z. Xu, Performance analysis of minimum variance CDMA receivers, *IEEE Trans. Signal Processing* 46 (11) (November 1998) 3014–3022.
- [17] M.K. Tsatsanis, Z. Xu, Constrained optimization methods for direct blind equalization, *IEEE Journal on Selected Areas in Commun.*, 1998 (accepted); see also 31 *Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, 2–5 November 1997.
- [18] B.D. Van Veen, K.M. Buckley, Beamforming: A versatile approach to spatial filtering, *IEEE Acoust. Speech Signal Process. Mag.* 5 (April 1998) 4–24.
- [19] B.D. Van Veen, Minimum variance beamforming, in: S. Haykin, A. Steinhardt (Eds.), *Adaptive Radar Detection and Estimation*, Wiley, New York, 1992.
- [20] X. Wang, H.V. Poor, Blind multiuser detection: A subspace approach, *IEEE Trans. Inform. Theory* 44 (2) (March 1998) 677–690.