Blind Constant Modulus Equalization via Convex Optimization

Boris Maričić, Zhi-Quan Luo, *Senior Member, IEEE*, and Timothy N. Davidson, *Member, IEEE*

***Abstract*—In this paper, we formulate the problem of blind equalization of constant modulus (CM) signals as a convex optimization problem. The convex formulation is obtained by performing an algebraic transformation on the direct formulation of the CM equalization problem. Using this transformation, the original nonconvex CM equalization formulation is turned into a convex semidefinite program (SDP) that can be efficiently solved using interior point methods. Our SDP formulation is applicable to baud spaced equalization as well as fractionally spaced equalization. Performance analysis shows that the expected distance between the equalizer obtained by the SDP approach and the optimal equalizer in the noise-free case converges to zero exponentially as the signal-to-noise ratio (SNR) increases. In addition, simulations suggest that our method performs better than standard methods while requiring significantly fewer data samples.**

***Index Terms*—Blind equalization, constant modulus, convex optimization.**

I. Introduction

CONVENTIONAL equalization and carrier recovery algorithms generally require an initial training period during which a known data sequence is transmitted over the channel and synchronized at the receiver. In applications involving the transmission of long data packets over a time-invariant channel, the consequent reduction in the data throughput of the system is negligible because only one training period is required. However, in applications in which the data packets are short or the time-variation of the channel is significant, such as in distributed networks and mobile systems, training can be rather inefficient, particularly at low signal-to-noise ratios (SNRs) [1]. In such applications, it may be preferable to equalize the communication channel in an unsupervised (blind) manner. The essence of blind equalization rests on the exploitation of structure of the channel and/or the properties of the input. For man-made signals, such as those encountered in wireless communications, the signal properties are often well known. Many digital communications schemes involve the transmission of *constant modulus* (CM) signals; hence, several schemes for blind equalization of CM signals have been developed [2]. Typically, they are based on gradient descent minimization of a specially designed cost function [3]–[5]. However, due to the nonconvexity of the cost functions, these gradient descent-based algorithms can experience undesirable local convergence problems, which may result in insufficient removal of channel distortion [2], [6]–[10]. In fact, depending on the initialization and the choice of stepsize, these algorithms suffer local minima and slow convergence. Unfortunately, no suitable default initial points or stepsizes are known.

In this paper, we formulate the problem of blind equalization of CM signals as a convex optimization problem. We first consider the case of baud spaced equalization of quadrature phase shift keying (QPSK) symbols transmitted through a complex-valued channel. We perform an algebraic transformation on the direct formulation of the CM equalization problem. As a result of this transformation, the original nonconvex CM equalization formulation is converted into a convex *semidefinite program* (SDP). Semidefinite programs consist of a linear (convex) objective function and (convex) linear matrix inequality constraints and can be efficiently solved using interior point methods [11]. We then present a natural extension of our SDP method to the class of *Fractionally spaced equalizers* (FSEs).

There are two major advantages of convex formulations of CM equalization. First, they do not suffer from local minima, and there is a well-developed theory of algorithm initialization and the choice of stepsize. This makes the convex optimization approach to blind equalization robust to the uncertainties of practical communication environments. As a result, our convex optimization-based blind CM equalization algorithms require far fewer data samples than the standard blind adaptive equalization methods in [3]–[5], which are not globally convergent [2],[6], [8]. The second major advantage of our convex formulation is that there are highly efficient algorithms for its solution. Our simulation results indicate that our convex approach performs better than the standard methods, even when we use some *a priori* knowledge of the channel impulse response to aid initialization of the standard methods of [3]–[5]. In addition, our simulations confirm that the convex approach requires significantly fewer samples, thus making it attractive in applications where convergence times requiring thousands of input samples are undesirable. Our simulations also indicate that our formulation is more reliable than the *analytical constant modulus algorithm* (ACMA) [12], [13].

This paper is organized as follows. In Section II, we introduce blind equalization of CM signals. In Section III, we develop the convex formulation of baud spaced blind CM equalization for the case of a QPSK signal transmitted through a complex-valued channel. Section IV contains a performance analysis of our algorithm. In Section V, we extend the framework to the class of

Manuscript received June 11, 2001; revised October 25, 2002. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant OPG0090391. The second author was also supported by the Canada Research Chairs program. The associate editor coordinating the review of this paper and approving it for publication was Dr. Chong-Yung Chi.

The authors are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, L8S 4K1 Canada (e-mail: maricib@mcmaster.ca; davidson@mcmaster.ca; luozq@mcmaster.ca).

Digital Object Identifier 10.1109/TSP.2002.808112

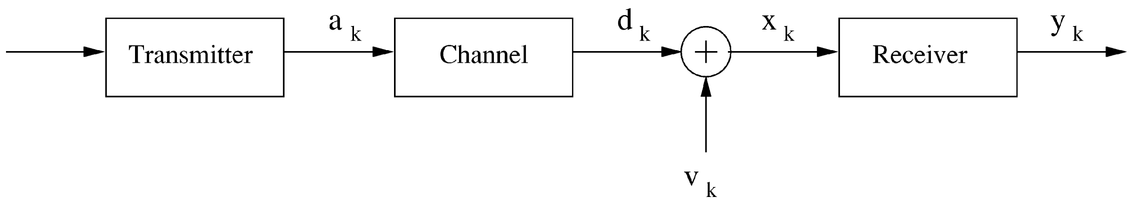


Fig. 1. Block diagram of the system.

fractionally spaced equalizers, and in Section VI, we present our simulation results. Section VII contains some concluding remarks. In Appendix A, we give a brief overview of convexity, semidefinite programming and the interior-point methods that are used to efficiently solve convex optimization problems. Appendix B contains a detailed analysis of the perturbation theory used in Section IV.

A few remarks on our notations are in order. All vectors are in boldfaced lowercase letters. Boldfaced upper case letters are used to denote matrices. Matrix transposition is denoted by a superscript ^{T}, whereas the Hermitian transposition for a complex matrix is represented by the superscript ^{H}. Moreover, the notation $\mathbf{G} \succeq \mathbf{0}$ signifies the semidefiniteness ($\mathbf{x}^H\mathbf{G}\mathbf{x} \geq 0,\ \forall \mathbf{x}$) of a (Hermitian) symmetric matrix $\mathbf{G}$. The matrix inner product is defined as $\mathbf{A} \bullet \mathbf{B} = \sum_{j,k} A^*_{jk} B_{jk}$, where $\mathbf{A}, \mathbf{B}$ are two square matrices of the same dimension, and the superscript * denotes complex conjugation.

II. Blind CM Equalization

Consider the digital communication system shown in Fig. 1, where a_k represents the CM input signal, and v_k is additive white Gaussian noise. The output of the equalizer y_k can be expressed as

$$y_k = \mathbf{w}^H \mathbf{x}_k \tag{2.1}$$

where $\mathbf{x}_k = [x_k, x_{k-1}, \ldots, x_{k-n+1}]^T$, $\mathbf{w}$ is an n-dimensional (complex) weight vector, and n is the length of the equalizer. If perfect equalization is achieved, the sequence y_k will be of the CM type, just like the input signal a_k. Consequently, a natural formulation of the CM equalization problem is [3]

$$\text{minimize} \quad \frac{1}{N}\sum_{k=1}^{N}\left(|y_k|^2 - 1\right)^2 \tag{2.2}$$

where N is the length of the sequence y_k. Here, we have assumed that the magnitude of the CM signal is equal to one, but there is no loss of generality in this assumption since a different magnitude will only result in a rescaling of the optimal equalizer. Using (2.1) and (2.2), we can rewrite the CM equalization problem as

$$\text{minimize} \quad \frac{1}{N}\sum_{k=1}^{N}\left(\left|\mathbf{w}^H\mathbf{x}_k\right|^2 - 1\right)^2. \tag{2.3}$$

Since

$$\left|\mathbf{w}^H\mathbf{x}_k\right|^2 = \mathbf{w}^H\mathbf{x}_k\left(\mathbf{w}^H\mathbf{x}_k\right)^H = \mathbf{w}^H\mathbf{x}_k\mathbf{x}_k^H\mathbf{w}$$

we have the following optimization problem:

$$\text{minimize} \quad \frac{1}{N}\sum_{k=1}^{N}\left(\mathbf{w}^H\mathbf{X}_k\mathbf{w} - 1\right)^2 \tag{2.4}$$

where $\mathbf{X}_k = \mathbf{x}_k\mathbf{x}_k^H$.

Various gradient descent algorithms have been proposed to solve the minimization problem in (2.4). However, the objective function in (2.4) is a fourth-order multivariate polynomial that is nonconvex. This makes the gradient based algorithms vulnerable to the traps of local minima, as has been well recognized in the literature [7], [8]. In the subsequent sections, we will show how (2.4) can be reformulated as a convex optimization problem. In particular, the problem can be formulated as the minimization of a linear objective function subject to linear and semidefiniteness constraints, that is, as an SDP. SDPs can be efficiently solved using interior point methods [11].

III. Convex Reformulation: Baud Spaced Equalizers

In this section, we present our SDP reformulation for the baud spaced blind equalization of QPSK symbols transmitted through a complex-valued channel. The framework can be extended directly to fractionally spaced equalization, as will be shown in Section V.

A. SDP Formulation

From (2.4), we see that our optimization problem can be written as

$$\text{minimize} \quad f(\mathbf{w}) = \frac{1}{N}\sum_{k=1}^{N}\left(\mathbf{w}^H\mathbf{X}_k\mathbf{w} - 1\right)^2. \tag{3.1}$$

Notice that $f(\mathbf{w})$ is a fourth-order polynomial in $\mathbf{w}$, which is a complex-valued n-dimensional vector. It is important to note that although $\mathbf{w}$ is a vector containing complex entries, $f(\mathbf{w})$ is always real-valued (since $\mathbf{X}_k$ is Hermitian). The above optimization problem can be recast as

$$\begin{aligned}&\text{maximize } \tau\\ &\text{subject to } f(\mathbf{w}) - \tau \geq 0, \quad \text{for all } \ \mathbf{w}.\end{aligned} \tag{3.2}$$

We can think of τ as a horizontal hyperplane that lies beneath $f(\mathbf{w})$ for every value of $\mathbf{w}$. Instead of minimizing $f(\mathbf{w})$, we lift up (maximize) the hyperplane while requiring it to always lie below $f(\mathbf{w})$. At the optimal solution, τ_{opt} equals to the optimal (minimum) value of $f(\mathbf{w})$. The formulation in (3.2) has a linear objective and linear constraints in τ; hence, it is convex. However, (3.2) remains difficult to solve because the constraint is infinite dimensional, i.e., there is one constraint for each $\mathbf{w} \in \mathbb{C}^n$.

Furthermore, extraction of a corresponding equalizer (i.e., a $\mathbf{w}$ such that $f(\mathbf{w}) = \tau_{opt}$) can be problematic. In this section, we derive an *efficiently solvable* convex formulation of (a restriction of) the problem in (3.2), and in Section III-B, we will determine an efficient post-processing procedure to determine the corresponding equalizer. Performance issues are discussed in Sections IV and VI.

In order to transform (2.4) into a convex problem, we first define two convex cones of real-valued fourth-order polynomials of $\mathbf{w}$, $\mathcal{C}$ and $\mathcal{D}$:

$$\mathcal{C} = \Big\{ g \mid g(\mathbf{w}) \text{ is a real valued fourth-order polynomial of } \mathbf{w} \text{ and } g(\mathbf{w}) \geq 0, \ \forall\, \mathbf{w} \Big\}$$

$$\mathcal{D} = \Big\{ g \mid g(\mathbf{w}) = \sum_i g_i(\mathbf{w})^2, \text{ with each } g_i(\mathbf{w}) \text{ being a real-valued second-order polynomial of } \mathbf{w} \Big\}.$$

In other words, $\mathcal{C}$ is the convex cone of nonnegative real-valued polynomials of degree 4, and $\mathcal{D}$ is the convex cone of all fourth-order polynomials that can be represented as the sum of squares of some real-valued quadratic polynomials. Obviously, $\mathcal{D}$ is a subset of $\mathcal{C}$, i.e., $\mathcal{D} \subseteq \mathcal{C}$. We can express our optimization problem (3.2) in terms of $\mathcal{C}$ as follows:

$$\begin{aligned} &\text{maximize } \tau \\ &\text{subject to } f(\mathbf{w}) - \tau \in \mathcal{C}. \end{aligned} \tag{3.3}$$

Now consider a "restriction" of (3.3) in which the feasible set is constrained so that $f(\mathbf{w}) - \tau$ lies in the cone $\mathcal{D}$:

$$\begin{aligned} &\text{maximize } \tau \\ &\text{subject to } f(\mathbf{w}) - \tau \in \mathcal{D}. \end{aligned} \tag{3.4}$$

Let $\bar{\tau}_{opt}$ be the optimal value of (3.4). In general, the optimization problem (3.4) is not necessarily equivalent to the one in (3.3) since the latter restricts the feasible set to a subset of the original feasible set. [The formulation (3.4) will later be shown to be a convex semi-definite programming problem, hence, efficiently solvable.] Since (3.4) is a restriction of (3.3), it provides a lower bound on the optimal value of τ_{opt} in (3.3), i.e., $\tau_{opt} \geq \bar{\tau}_{opt}$. From the definition of $\mathcal{D}$, it is also clear that $\bar{\tau}_{opt} \geq 0$. If the optimal equalizer $\mathbf{w}_{opt}$ of (3.3) satisfies $f(\mathbf{w}_{opt}) = 0$ (as is the case under certain ideal conditions), then the optimal value τ_{opt} of (3.3) is zero. In this case, we find that the optimal value $\bar{\tau}_{opt}$ of (3.4) is also zero. In other words, we have the following proposition.

Proposition III.1: If perfect constant modulus equalization is achieved in (3.3) by the equalizer $\mathbf{w}_{opt}$ in the sense that $\tau_{opt} = f(\mathbf{w}_{opt}) = 0$, then the optimal equalizer obtained from the formulation (3.4) also achieves perfect constant equalization with the optimal value $\bar{\tau}_{opt} = 0$.

Proof: Although a proof can be simply derived for the inequality $\tau_{opt} \geq \bar{\tau}_{opt} \geq 0$, the following proof illustrates the structure of the restriction in (3.4). Let $g(\mathbf{w};\tau) = f(\mathbf{w}) - \tau = (1/N)\sum_{k=1}^{N}(|\mathbf{w}^H\mathbf{x}_k|^2 - 1)^2 - \tau$. If perfect constant modulus equalization is achieved with $\mathbf{w}_{opt}$, then we have $|y_k| = |\mathbf{w}_{opt}^H\mathbf{x}_k| = 1$, for all k. Therefore, $f(\mathbf{w}_{opt}) = 0$, and $\tau_{opt} = 0$. We can then write

$$g(\mathbf{w};\tau_{opt}) = f(\mathbf{w}) - \tau_{opt} = \frac{1}{N}\sum_{k=1}^{N}\left(\left|\mathbf{w}^H\mathbf{x}_k\right|^2 - 1\right)^2$$

which is clearly in the form of sum of squares of quadratic polynomials. Therefore, $g(\mathbf{w};\tau_{opt}) \in \mathcal{D}$, implying that $\tau_{opt} = 0$ is a feasible point of the formulation (3.4). Consequently, perfect constant equalization is achievable through the formulation in (3.4), and hence, $\bar{\tau}_{opt} = 0$. Q.E.D.

It has been shown [23] that under the spatio-temporal diversity assumption and in the absence of channel noise, linear equalizers can achieve perfect equalization. In this case, according to Proposition 3.1, the restricted set $\mathcal{D}$ retains the optimal equalizer. In other words, in the noise-free case and with spatio-temporal diversity, solving the optimization problem (3.4) will yield perfect CM equalization. In the absence of these assumptions, we can still use the formulation (3.4) (instead of the formulation (3.3)) for blind CM equalization, with the expectation that the presence of noise will only mildly perturb the optimal equalizer. This is indeed the case, as we will show later in Section IV.

An advantage of formulation (3.4) is that it can be converted into a (convex) semi-definite programming problem that can be solved via highly efficient interior point algorithms. This is what we will show next. To perform this transformation, we need the following lemma, which is a special case of a more general result of Nesterov [14].

Lemma III.1: Given any fourth-order polynomial $g(\mathbf{w})$, the following relation holds:

$$\begin{aligned} &g(\mathbf{w}) \in \mathcal{D} \iff g(\mathbf{w}) = \bar{\mathbf{w}}^H\mathbf{G}\bar{\mathbf{w}} \\ &\text{for some Hermitian matrix } \mathbf{G} \succeq \mathbf{0} \end{aligned} \tag{3.5}$$

where

$$\bar{\mathbf{w}} = \left(\mathbf{u}_{(2)}^H, \mathbf{u}_{(1)}^H, \mathbf{u}_{(0)}^H\right)^H \in \mathbb{C}^{2n^2+3n+1} \tag{3.6}$$

with $\mathbf{u} = (\mathbf{w}^T, \mathbf{w}^H)^T$ and $\mathbf{u}_{(2)}$ and $\mathbf{u}_{(1)}$ being vectors whose components are the products $\{u_iu_j,\ 1 \leq i \leq j \leq 2n\}$ and $\{u_i\}$, respectively, arranged in the usual lexicographic order, and $\mathbf{u}_{(0)} = 1$.

Proof: First, we notice that the components of the vector $\bar{\mathbf{w}}$ form a basis for quadratic polynomials in $\mathbf{w}$. Since the cost function $f(\mathbf{w})$ in (3.1) is a fourth-order polynomial that does not have the third- and the first-order part, it is possible to specialize Lemma III.1 to the case at hand and omit $\mathbf{u}_{(1)}$ in the definition of $\bar{\mathbf{w}}$. This will, in turn, decrease the complexity of our algorithm. However, we decided to include $\mathbf{u}_{(1)}$ in our analysis since it allows our algorithm to be applied to more general scenarios, i.e., when the cost function is an arbitrary fourth-order polynomial.

By definition, the set $\mathcal{D}$ contains all fourth-order polynomial functions that can be written as

$$g(\mathbf{w}) = \sum_i g_i^2(\mathbf{w}). \tag{3.7}$$

Since each $g_i(\mathbf{w})$ is a quadratic polynomial function in $\mathbf{w}$, we can represent $g_i(\mathbf{w})$ using the basis $\bar{\mathbf{w}}$ in the form

$g_i(\mathbf{w}) = \bar{\mathbf{w}}^H \mathbf{q}_i$, where $\mathbf{q}_i \in \mathbb{C}^{2n^2+3n+1}$ is a vector of coefficients. Hence, we have

$$g_i^2(\mathbf{w}) = \bar{\mathbf{w}}^H \mathbf{q}_i \mathbf{q}_i^H \bar{\mathbf{w}}$$

which further implies

$$\begin{aligned} g(\mathbf{w}) &= \bar{\mathbf{w}}^H \sum_i \left(\mathbf{q}_i \mathbf{q}_i^H\right) \bar{\mathbf{w}} \\ &= \bar{\mathbf{w}}^H \mathbf{G} \bar{\mathbf{w}}, \text{with } \mathbf{G} = \sum_i \left(\mathbf{q}_i \mathbf{q}_i^H\right) \succeq \mathbf{0}. \end{aligned} \tag{3.8}$$

Conversely, if (3.8) holds for some Hermitian positive semidefinite matrix $\mathbf{G}$, then we can factorize $\mathbf{G}$ as the sum of Hermitian rank one matrices, i.e., $\mathbf{G} = \sum_i (\mathbf{q}_i \mathbf{q}_i^H)$. (For example, the eigen-decomposition of $\mathbf{G}$ generates such a factorization.) Now, if we define the quadratic polynomials $g_i(\mathbf{w}) = \mathbf{q}_i^H \bar{\mathbf{w}}$, then we have (3.7), as desired. This completes the proof of the lemma. Q.E.D.

Notation: In the subsequent development, it will be convenient to denote the entries of $\mathbf{G}$ according to the ordering of $\bar{\mathbf{w}}$. We partition the matrix $\mathbf{G}$ as

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{22} & \mathbf{G}_{21} & \mathbf{G}_{20} \\ \mathbf{G}_{21}^H & \mathbf{G}_{11} & \mathbf{G}_{10} \\ \mathbf{G}_{20}^H & \mathbf{G}_{10}^H & \mathbf{G}_{00} \end{bmatrix} \tag{3.9}$$

where $\mathbf{G}_{ij}$ corresponds to the coefficients for the product terms between entries of $\mathbf{u}_{(i)}$ and $\mathbf{u}_{(j)}$. In particular, the submatrices $\mathbf{G}_{20}, \mathbf{G}_{10}$ are vectors, and the submatrix $\mathbf{G}_{00}$ is actually a scalar because it corresponds to the product $\mathbf{u}_{(0)} = 1$ with itself, which is the constant term. We will further denote the entries of $\mathbf{G}_{22}$ by $\mathbf{G}_{(ij)(\ell m)}$ since the notation naturally signifies the coefficient for the product term between entries $u_i u_j$ and $u_\ell u_m$ in $\mathbf{u}_{(2)}$, where $i \le j, \ell \le m$. Similarly, we use $\mathbf{G}_{(ij)(\ell)}$ and $\mathbf{G}_{(i)(j)}$ to denote the entries of $\mathbf{G}_{21}$ and $\mathbf{G}_{11}$. Since $\mathbf{u}_{(0)} = 1$, the entries of $\mathbf{G}_{20}$ and $\mathbf{G}_{10}$ will be denoted by $\mathbf{G}_{(ij)}$ and $\mathbf{G}_{(\ell)}$, respectively.

Lemma III.1 implies that the convex cone $\mathcal{D}$ can be equivalently characterized by

$$\mathcal{D} = \left\{ g(\mathbf{w}) | g(\mathbf{w}) = \bar{\mathbf{w}}^H \mathbf{G} \bar{\mathbf{w}} \text{ for some } \mathbf{G} \succeq \mathbf{0}. \right\}.$$

Using this characterization in (3.4), we obtain an alternative formulation of (3.4) as

$$\begin{aligned} &\text{maximize} \quad \tau \\ &\text{subject to } \frac{1}{N} \sum_{k=1}^{N} \left(\mathbf{w}^H \mathbf{X}_k \mathbf{w} - 1\right)^2 - \tau = \bar{\mathbf{w}}^H \mathbf{G} \bar{\mathbf{w}}, \text{ for all } \mathbf{w} \\ &\quad \text{and } \mathbf{G} \succeq \mathbf{0}. \end{aligned} \tag{3.10}$$

Notice that the constraint

$$\frac{1}{N} \sum_{k=1}^{N} \left(\mathbf{w}^H \mathbf{X}_k \mathbf{w} - 1\right)^2 - \tau = \bar{\mathbf{w}}^H \mathbf{G} \bar{\mathbf{w}}, \quad \text{for all } \mathbf{w} \tag{3.11}$$

signifies that the two fourth-order polynomials are identical. This is possible if and only if the corresponding coefficient vectors, when expressed over an appropriate basis of the fourth-order polynomials, are equal. We will use this property to relate the entries of $\mathbf{G}$ to the data matrices $\mathbf{X}_k$. In particular, notice that the components of the vector

$$\tilde{\mathbf{w}} = \left(\mathbf{u}_{(4)}^H, \mathbf{u}_{(3)}^H, \mathbf{u}_{(2)}^H, \mathbf{u}_{(1)}^H, 1\right)^H \tag{3.12}$$

form a basis of the vector space of all fourth-order complex coefficient polynomials of $\mathbf{w}$, where $\mathbf{u}_{(4)}$, $\mathbf{u}_{(3)}$, $\mathbf{u}_{(2)}$, and $\mathbf{u}_{(1)}$ are vectors whose components are given by the products $\{u_i u_j u_\ell u_m, \ 1 \le i \le j \le \ell \le m \le 2n\}$, $\{u_i u_j u_\ell, \ 1 \le i \le j \le \ell \le 2n\}$, $\{u_i u_j, \ 1 \le i \le j \le 2n\}$ and $\{u_i\}$, respectively, arranged in the usual lexicographic order and, as previously defined, $\mathbf{u} = (\mathbf{w}^T, \mathbf{w}^H)^T$. Thus, an arbitrary fourth-order polynomial of $\mathbf{w}$, $f(\mathbf{w})$ can be written as $f(\mathbf{w}) = \mathbf{p}^H \tilde{\mathbf{w}}$ for some coefficient vector $\mathbf{p}$.

To simplify the constraint (3.11), we represent the left- and the right-hand side polynomials using the basis vector $\tilde{\mathbf{w}}$. Let us define the sample fourth-order cumulant of the channel output sequence $\{x_k\}$ as

$$\text{Cum}(i,j,\ell,m) = \frac{1}{N} \sum_{k=1}^{N} x'_{k-i+1} x'_{k-j+1} x'_{k-\ell+1} x'_{k-m+1} \qquad \forall\, 1 \le i,j,\ell,m \le 2n$$

where $x'_{k-i+1} = x^*_{k-i+1}$ for $1 \le i \le n$, and $x'_{k-i+1} = x_{k-i+1}$ otherwise (* denotes complex conjugation). For finite N, the above expression involves data samples x_k whose indices are out of range; when this happens, we always assume $x_k = 0$. We also define the sample covariance matrix of the (zero mean) sequence $\mathbf{x}'_k = \left(\mathbf{x}_k^H, \mathbf{x}_k^T\right)^T$, where $\mathbf{x}_k = [x_k, x_{k-1}, \ldots, x_{k-n+1}]^T$ as in (2.1), as

$$\mathbf{X} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{X}_k. \tag{3.13}$$

Moreover, for all $1 \le i \le j \le \ell \le m \le 2n$, we let $\mathcal{P}_{i,j,\ell,m}$ denote the set of all *distinct* 4-tuples that are permutations of (i,j,ℓ,m). For example

$$\mathcal{P}_{1,1,2,2} = \{(1,1,2,2),\ (1,2,1,2),\ (1,2,2,1) \quad (2,2,1,1),\ (2,1,2,1),\ (2,1,1,2)\}.$$

We further define a subset $\mathcal{Q}_{i,j,\ell,m}$ of $\mathcal{P}_{i,j,\ell,m}$:

$$\mathcal{Q}_{i,j,\ell,m} = \{(i',j',\ell',m') \mid (i',j',\ell',m') \in \mathcal{P}_{i,j,\ell,m} \text{ and } i' \le j',\ \ell' \le m'\}. \tag{3.14}$$

Similarly, for $1 \le i \le j \le \ell \le 2n$, we let $\mathcal{P}_{i,j,\ell}$ (respectively, $\mathcal{P}_{i,j}$) denote the set of distinct triples (respectively, pairs) that are permutations of (i,j,ℓ) [respectively, (i,j)]. Then, we define the index set

$$\mathcal{Q}_{i,j,\ell} = \left\{ (i',j',\ell') \mid (i',j',\ell') \in \mathcal{P}_{i,j,\ell} \text{ and } i' \le j' \right\}. \tag{3.15}$$

For example, we have

$$\begin{aligned} \mathcal{Q}_{1,1,2,2} &= \{(1,1,2,2),(1,2,1,2),(2,2,1,1)\} \\ \mathcal{Q}_{1,1,2} &= \{(1,1,2),(1,2,1)\}. \end{aligned}$$

Now, we can define the symmetric fourth-order cumulant as

$$C_{i,j,\ell,m} = \sum_{(i',j',\ell',m')\in\mathcal{P}_{i,j,\ell,m}} \text{Cum}\,(i',j',\ell',m'). \quad (3.16)$$

Then, we have

$$\frac{1}{N}\sum_{k=1}^{N}\left(\mathbf{w}^H\mathbf{X}_k\mathbf{w}-1\right)^2-\tau = \sum_{1\le i\le j\le\ell\le m\le 2n} C_{i,j,\ell,m}u_iu_ju_\ell u_m - 2\sum_{1\le i\le 2n}\mathbf{X}_{ii}u_i^2 - 4\sum_{1\le i<j\le 2n}\mathbf{X}_{ij}u_iu_j + 1-\tau \quad (3.17)$$

where $\mathbf{X}_{ij}$ is the (i,j)th element of the sample covariance matrix $\mathbf{X}$ in (3.13). On the other hand, using the notation $\mathcal{Q}_{i,j,\ell,m}$, we can rewrite the polynomial $\bar{\mathbf{w}}^H\mathbf{G}\bar{\mathbf{w}}$ as

$$\begin{aligned}\bar{\mathbf{w}}^H\mathbf{G}\bar{\mathbf{w}} = &\sum_{1\le i\le j\le\ell\le m\le 2n}\left(\sum_{(i',j',\ell',m')\in\mathcal{Q}_{i,j,\ell,m}}\mathbf{G}_{(i'j')(\ell',m')}\right)u_iu_ju_\ell u_m\\ &+\sum_{1\le i\le j\le\ell\le 2n}\left(\sum_{(i',j',\ell')\in\mathcal{Q}_{i,j,\ell}}\mathbf{G}_{(i'j')(\ell')}\right)u_iu_ju_\ell\\ &+\sum_{1\le i\le j\le 2n}\left(2\mathbf{G}_{(ij)}+(2-\delta(i-j))\mathbf{G}_{(i)(j)}\right)u_iu_j\\ &+\sum_{1\le i\le 2n}\mathbf{G}_{(i)}u_i+\mathbf{G}_{00}.\end{aligned} \quad (3.18)$$

Now, we have represented the left- and right-hand side polynomials in (3.11) over the basis $\tilde{\mathbf{w}}$ [see (3.17) and (3.18)]. We can compare the coefficients of both representations to obtain the set of linear constraints on $\mathbf{G}$ in (3.19), shown at the bottom of the page, where $\delta(\cdot)$ is the usual Kronecker delta function, the notations $\mathbf{G}_{(ij)(\ell m)}$, $\mathbf{G}_{(ij)(\ell)}$, $\mathbf{G}_{(ij)}$, $\mathbf{G}_{(i)(j)}$, $\mathbf{G}_{(i)}$ and $\mathbf{G}_{00}$ were defined after (3.9) and the index sets $\mathcal{Q}_{i,j,\ell,m}$ and $\mathcal{Q}_{i,j,\ell}$ are defined in (3.14) and (3.15).

Finally, we can use (3.19) to replace the constraint in (3.10) and obtain the following equivalent formulation:

$$\begin{aligned}&\text{maximize } \tau\\ &\text{subject to } \mathbf{G} \text{ satisfies (3.19)}\\ &\qquad\mathbf{G}\succeq\mathbf{0}.\end{aligned} \quad (3.20)$$

Problem (3.20) has a linear objective function, and the variables τ and $\mathbf{G}$ are subject to linear equality constraints and a linear matrix inequality constraint. Hence, (3.20) is an SDP [albeit in a mildly nonstandard form] and can be efficiently solved using interior point methods [11]. (We have used the SeDuMi implementation [15].) By substituting the constraint $\mathbf{G}_{00}=1-\tau$ into the objective function to remove τ, the linear objective function becomes equal to $-\mathbf{G}_{00}$, and the problem can be formulated in the standard form of a SDP problem:

$$\begin{aligned}&\text{maximize } \mathbf{C}\bullet\mathbf{G}\\ &\text{subject to } \mathbf{A}_i\bullet\mathbf{G}=\mathbf{p}_i,\quad i=1,\ldots,m\\ &\qquad\mathbf{G}\succeq\mathbf{0}\end{aligned} \quad (3.21)$$

where $\mathbf{C}$ is a sparse matrix with only one nonzero element, and $\mathbf{A}_i\bullet\mathbf{G}=\mathbf{p}_i$, $i=1,\ldots,m$ denote all the linear constraints in (3.19) except the last one $\mathbf{G}_{00}=1-\tau$. In the next section, we will show how to find an optimal equalizer $\mathbf{w}$ corresponding to the optimal solution of the SDP in (3.20) [or, equivalently, (3.21)].

B. Post-Processing

The solution of (3.20) provides the optimal value of $\mathbf{G}$, which we denote by $\mathbf{G}_{opt}$. It remains to determine an optimal equalizer $\mathbf{w}_{opt}$ from $\mathbf{G}_{opt}$. To this end, we note that $\mathbf{u}=(\mathbf{w}^T,\mathbf{w}^H)^T$ is contained in $\bar{\mathbf{w}}$ (3.6); thus, once we have $\bar{\mathbf{w}}$, it is straightforward to obtain $\mathbf{u}$ and ultimately $\mathbf{w}$. Therefore, we will endeavor to find $\bar{\mathbf{w}}_{opt}\in\mathbb{C}^{2n^2+3n+1}$ first. According to Proposition III.1, if perfect constant modulus equalization is achieved, then there is an optimal vector $\bar{\mathbf{w}}_{opt}$ such that

$$f(\mathbf{w}_{opt})=\bar{\mathbf{w}}_{opt}^H\mathbf{G}_{opt}\bar{\mathbf{w}}_{opt}=0. \quad (3.22)$$

Since $\mathbf{G}_{opt}$ is positive semidefinite, it follows from (3.22) that $\bar{\mathbf{w}}_{opt}$ lies in the null space of $\mathbf{G}_{opt}$. If the null space of $\mathbf{G}_{opt}$ has dimension 1, then we can determine $\bar{\mathbf{w}}_{opt}$ uniquely (up to a scaling factor). When perfect constant equalization is not achievable, then we should look for the eigenspace $\mathcal{N}(\mathbf{G}_{opt})$ corresponding to the *almost* zero eigenvalues of $\mathbf{G}_{opt}$. Again, if $\mathcal{N}(\mathbf{G}_{opt})$ has dimension 1, then $\bar{\mathbf{w}}_{opt}$ can be computed uniquely (up to a scaling). When the dimensionality of $\mathcal{N}(\mathbf{G}_{opt})$ is greater than 1, then we must look among the vectors in $\mathcal{N}(\mathbf{G}_{opt})$ for a vector that has the specific structure defined by (3.6).

Let $\mathcal{M}\subset\mathbb{C}^{2n^2+3n+1}$ denote the set of vectors $\bar{\mathbf{w}}$ of the form

$$\bar{\mathbf{w}}=\left(\mathbf{u}_{(2)}^H,u_0\mathbf{u}_{(1)}^H,u_0^2\right)^H \quad (3.23)$$

where $\mathbf{u}_{(2)}$ and $\mathbf{u}_{(1)}$ are vectors whose components are the products $\{u_iu_j,\ 1\le i\le j\le 2n\}$ and $\{u_i,\ 1\le i\le 2n\}$,

$$\begin{cases}\displaystyle\sum_{(i',j',\ell',m')\in\mathcal{Q}_{i,j,\ell,m}}\mathbf{G}_{(i'j')(\ell',m')}=C_{i,j,\ell,m}, & 1\le i\le j\le\ell\le m\le 2n\\ \displaystyle\sum_{(i,j',\ell')\in\mathcal{Q}_{i,j,\ell}}\mathbf{G}_{(i'j')(\ell')}=0, & 1\le i\le j\le\ell\le 2n\\ 2\mathbf{G}_{(ij)}+(2-\delta(i-j))\mathbf{G}_{(i)(j)}=-2(2-\delta(i-j))\mathbf{X}_{ij}, & 1\le i\le j\le 2n\\ \mathbf{G}_{(i)}=0, & 1\le i\le 2n\\ \mathbf{G}_{00}=1-\tau\end{cases} \quad (3.19)$$

respectively, arranged in the usual lexicographic order. Comparing (3.23) and (3.6), we see that a vector $\bar{\mathbf{w}}$ is of the form (3.6) for some $\mathbf{u} = (\mathbf{w}^T, \mathbf{w}^H)^T \in \mathbb{C}^{2n}$ if and only if $\bar{\mathbf{w}} \in \mathcal{M}$ and $u_0 = 1$. Since $\bar{\mathbf{w}}_{opt}$ must be of the form (3.6), we know $\bar{\mathbf{w}}_{opt} \in \mathcal{M} \cap \mathcal{H}$, where $\mathcal{H}$ denotes the hyperplane in $\mathbb{C}^{2n^2+3n+1}$, which consist of all vectors whose last component is equal to 1.

To summarize, $\bar{\mathbf{w}}_{opt}$ lies in the intersection $\mathcal{N}(\mathbf{G}_{opt}) \cap \mathcal{H} \cap \mathcal{M}$. This motivates the use of an alternating projection algorithm to find $\bar{\mathbf{w}}_{opt}$. Before we present the algorithm, it is important to notice that any vector $\bar{\mathbf{w}}$ in $\mathcal{M}$ can be viewed naturally as a Hermitian rank one matrix of size $(2n+1) \times (2n+1)$. In particular, there is a one-to-one correspondence between the vector $\bar{\mathbf{w}}$ given by (3.23) and the matrix $\bar{\mathbf{W}} = \mathbf{a}\mathbf{a}^H$, where $\mathbf{a} = (u_1, u_2, \ldots, u_{2n}, u_0)^T$, that is

$$\bar{\mathbf{W}} = \begin{bmatrix} u_1^2 & u_1u_2 & \cdots & u_1u_{2n} & u_1u_0 \\ u_2u_1 & u_2^2 & \cdots & u_2u_{2n} & u_2u_0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{2n}u_1 & u_{2n}u_2 & \cdots & u_{2n}^2 & u_{2n}u_0 \\ u_0u_1 & u_0u_2 & \cdots & u_0u_{2n} & u_0^2 \end{bmatrix}. \quad (3.24)$$

In particular, each element in $\bar{\mathbf{w}}$ is mapped to a unique element in the upper triangle of the (Hermitian) matrix $\bar{\mathbf{W}}$. For an arbitrary vector $\bar{\mathbf{y}} \in \mathbb{C}^{2n^2+3n+1}$ (not necessarily in $\mathcal{M}$), we can use the *same* mapping to construct a Hermitian matrix $\bar{\mathbf{Y}}$. Obviously, $\bar{\mathbf{Y}}$ is not necessarily a rank-one matrix, unless $\bar{\mathbf{y}} \in \mathcal{M}$. Therefore, to project a given vector $\bar{\mathbf{y}} \in \mathbb{C}^{2n^2+3n+1}$ onto the set $\mathcal{M}$, we need only construct the matrix $\bar{\mathbf{Y}}$ and then perform a rank-one approximation of $\bar{\mathbf{Y}}$. The rank-one approximation of the Hermitian matrix $\bar{\mathbf{Y}}$ is given by $\lambda_{\max}\mathbf{c}\mathbf{c}^H$, where $\lambda_{\max}$ is the largest eigenvalue of $\bar{\mathbf{Y}}$, and $\mathbf{c}$ is the corresponding normalized eigenvector. If $\bar{\mathbf{z}}$ denotes the projection of $\bar{\mathbf{y}}$ onto $\mathcal{M}$, then $\bar{\mathbf{z}}$ can be obtained by arranging the components of the upper triangle of $\lambda_{\max}\mathbf{c}\mathbf{c}^H$ in the order implied by (3.23) and (3.24). Alternatively, one can recognize that if $\lambda_{\max}^{1/2}\mathbf{c}$ is partitioned as $[\mathbf{z}_{(1)}^H, z_0]^H$, where z_0 is scalar, then $\bar{\mathbf{z}}$ can be assembled using (3.23). We are now ready to state the algorithm.

`An Alternating Projection Algorithm for Computing` $\bar{\mathbf{w}}_{opt}$

Step 0: Initialization. Given $\mathbf{G}_{opt}$, select a small threshold $\gamma > 0$. Select an arbitrary vector $\bar{\mathbf{w}}^{(0)}$ (say the vectors of all ones) as the starting vector.

Step 1: Eigendecomposition. Given $\mathbf{G}_{opt}$, compute its eigen decomposition. Form the matrix $\mathbf{V}_t$ whose columns are the eigenvectors corresponding to eigenvalues of $\mathbf{G}_{opt}$ that are smaller than γ.

Step 2: Projection to $\mathcal{N}(\mathbf{G}_{opt}) \cap \mathcal{H}$. Given $\bar{\mathbf{w}}^{(k)}$ for some $k \geq 0$. Notice that $\mathcal{N}(\mathbf{G}_{opt}) \cap \mathcal{H}$ is an affine subspace *[27]*; therefore, the projection of $\bar{\mathbf{w}}^{(k)}$ onto $\mathcal{N}(\mathbf{G}_{opt}) \cap \mathcal{H}$ is straightforward. An even simpler (and equally effective) *approximate* projection procedure is to premultiply $\bar{\mathbf{w}}^{(k)}$ by the projection matrix $\mathbf{P} = \mathbf{V}_t\mathbf{V}_t^H$ and then rescale the resulting vector so that its last component becomes 1. Let the resulting vector be $\bar{\mathbf{y}}$.

Step 3: Projection to $\mathcal{M}$. Form the matrix $\bar{\mathbf{Y}}$ from the vector $\bar{\mathbf{y}}$ using the mapping described in the preceding paragraph. Compute the largest eigenvalue $\lambda_{\max}$ and the corresponding (normalized) eigenvector $\mathbf{c}$ of the symmetric matrix $\bar{\mathbf{Y}}$. Let $\begin{bmatrix} \mathbf{u}_{(1)}^{(k+1)} \\ u_0^{(k+1)} \end{bmatrix} = \lambda_{\max}^{1/2}\mathbf{c}$, where $u_0^{(k+1)}$ is scalar, and synthesize $\bar{\mathbf{w}}^{(k+1)}$ using *(3.23)*.

Step 4: Repeat. Return to Step 2 with $k := k+1$ until convergence.

IV. PERFORMANCE ANALYSIS

Proposition 3.1 showed that the restriction over the cone $\mathcal{D}$ retains the optimal equalizer whenever the perfect equalization can be achieved. The latter condition is known to be satisfied if noise is absent and spatio-temporal diversity exists [23]. (In Section V, we extend our convex SDP method to the class of fractionally spaced equalizers.) In the presence of channel noise however, the data vector $\mathbf{p}$ in (3.21) will be perturbed, which will in turn cause a perturbation of the optimal solution. In this section, we use an error analysis result of SDP [24] to bound the distance between the noise-free and noise-present optimal solution. Furthermore, we express the bound as a function of the SNR and the data block length N.

Lemma IV.1: Consider a system of mixed linear and positive semidefinite constraints:

$$\begin{cases} \mathbf{G} \in \mathbf{G_0} + \mathcal{A} \\ \mathbf{G} \in \mathcal{K} \end{cases} \quad (4.1)$$

where $\mathbf{G_0}$ is a given matrix in $\mathcal{B}^{m\times m}$ (the subspace of all $m \times m$ Hermitian matrices), $\mathcal{A}$ is a linear subspace of $\mathcal{B}^{m\times m}$, and $\mathcal{K}$ is the Hermitian positive semidefinite matrix cone. The distance from a given matrix to the feasible set in (4.1) is at most $O(\epsilon^{2^{-d}})$, where the constant in the big O notation is independent of $\mathbf{G_0}$, and ϵ is the amount of constraint violation defined as the sum of the distances to the cone $\mathcal{K}$ and the affine space $\mathbf{G_0} + \mathcal{A}$. The non-negative integer d is the so-called degree of singularity of the system and is bounded by

$$d \leq \min\left\{m-1, \dim\bar{\mathcal{A}}, \dim\bar{\mathcal{A}}^{\perp}\right\} \quad (4.2)$$

where m is the size of a positive semidefinite matrix in the cone $\mathcal{K}$ (in our case, $m = 2n^2 + 3n + 1$, with n being the equalizer length), and $\bar{\mathcal{A}}$ denotes the smallest linear subspace containing $\mathbf{G_0} + \mathcal{A}$, i.e.,

$$\bar{\mathcal{A}} = \left\{\mathbf{G} \in \mathcal{B}^{m\times m} \,|\, \mathbf{G} + t\mathbf{G_0} \in \mathcal{A} \quad \text{for some } t \in \mathbb{R}\right\}.$$

A detailed proof of Lemma IV.1 in the case of real symmetric matrices is given in [24], but the technique is easily extended to complex Hermitian matrices. Let us now apply Lemma IV.1 to our SDP formulation in (3.21). Notice that in the noise-free case, we have $\bar{\tau}_{opt} = 0$ so that we can write the optimal solution set of (3.21) as

$$S_{nf} = \left\{\bar{\mathbf{G}} \,|\, \mathbf{C} \bullet \bar{\mathbf{G}} \leq 0,\ \mathbf{A}_i \bullet \bar{\mathbf{G}} = \bar{\mathbf{p}}_i,\ \bar{\mathbf{G}} \succeq \mathbf{0}\right\} \quad (4.3)$$

where $\bar{\mathbf{p}}$ is the noise-free data vector. Similarly, in the presence of noise, we can write the solution set of (3.21) as

$$S_{np} = \{\mathbf{G} \mid \mathbf{C} \bullet \mathbf{G} \le \tau_{opt},\ \mathbf{A}_i \bullet \mathbf{G} = \mathbf{p}_i,\ \mathbf{G} \succeq \mathbf{0}\} \quad (4.4)$$

where $\mathbf{p}$ represents the data vector in the presence of noise, and $\tau_{opt} \ge 0$ is the corresponding optimal value of (3.21). Fix any optimal solution $\bar{\mathbf{G}}_{opt} \in S_{nf}$. By applying Lemma IV.1 to the system (4.4), we conclude that there exists an optimal solution $\mathbf{G}_{opt} \in S_{np}$ such that

$$\left\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\right\| = O\left(\epsilon^{2^{-(m-1)}}\right) \quad (4.5)$$

where ϵ denotes the amount of constraint violation of $\bar{\mathbf{G}}_{opt}$ in the linear matrix inequality system (4.4) defining S_{np}. Since $\bar{\mathbf{G}} \succeq \mathbf{0}$ and $\mathbf{C} \bullet \bar{\mathbf{G}}_{opt} \le 0 \le \tau_{opt}$, the only constraints in (4.4) violated by $\bar{\mathbf{G}}_{opt}$ are the linear constraints. Therefore, the constraint violation is $\epsilon = \|\bar{\mathbf{p}} - \mathbf{p}\|$. Notice that the constant associated with the big O notation in (4.5) is independent of $\mathbf{p}$ (thus independent of noise or data length). Consequently, we obtain

$$\left\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\right\| = O\left(\|\bar{\mathbf{p}} - \mathbf{p}\|^{2^{-(m-1)}}\right). \quad (4.6)$$

Since $\mathbf{p} \to \bar{\mathbf{p}}$ as the SNR and the data block length increase, the above bound (4.6) explains why our method is robust to the presence of some channel noise and the lack of perfect equalization, as will be shown by simulation in Section VI.

We now give an explicit estimate of ϵ as a function of the SNR and the data block length N. For simplicity, we consider the case of real channels and equalizers. In that case, $\mathbf{u} = \mathbf{w}$, i.e., $m = 2n + \binom{n}{2} + 1$. The complex case can be analyzed in a similar fashion. Notice that the cost function $f(\mathbf{w})$ in (3.1) is a fourth-order polynomial that does not have the third- and first-order parts, and the coefficients of $f(\mathbf{w})$ are given by the data vector $\mathbf{p}$. We can partition the data vector as $\mathbf{p} = (\mathbf{p}_{(4)}^T, \mathbf{p}_{(2)}^T, \mathbf{p}_{(0)}^T)^T$, where $\mathbf{p}_{(4)}$ and $\mathbf{p}_{(2)}$ contain polynomial coefficients corresponding to the fourth- and second-order parts, respectively, and $\mathbf{p}_{(0)} = 1$. Accordingly, we partition $\boldsymbol{\epsilon} = \bar{\mathbf{p}} - \mathbf{p}$ as $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_{(4)}^T, \boldsymbol{\epsilon}_{(2)}^T, \boldsymbol{\epsilon}_{(0)}^T)^T$. In Appendix B, we show that $E\left\{\boldsymbol{\epsilon}_{(4)}\right\} = O(\sigma_v^4 + \sigma_v^2)$, $E\left\{\boldsymbol{\epsilon}_{(2)}\right\} = O(\sigma_v^2)$, and $E\left\{\boldsymbol{\epsilon}_{(0)}\right\} = 0$, where σ_v^2 is the channel noise variance and is related to the ratio of the transmitted signal power to the received noise power (SNR) as

$$\sigma_v^2 = 10^{-(\mathrm{SNR}/10)} \quad (4.7)$$

assuming that the magnitude of the CM signal is equal to one. Therefore, we have that

$$E\{\epsilon\} = O(\sigma_v^4 + \sigma_v^2) \quad (4.8)$$

which implies that the expected value of the amount of constraint violation can be bounded as a function of noise power or, equivalently, as a function of the SNR.

Next, we need to compute $E\{\|\boldsymbol{\epsilon}\|\}$ since we will bound $E\{\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\|\}$ by $E\{\|\boldsymbol{\epsilon}\|^{2^{-(m-1)}}\}$. This can be done in a similar fashion as preceding analysis and is relegated to Appendix B. The final estimate is in the form

$$E\{\|\boldsymbol{\epsilon}\|\} = O(\sigma_v^4 + \sigma_v). \quad (4.9)$$

Recall that $m = 2n + \binom{n}{2} + 1$, where n is the equalizer length. The above estimate (4.9), together with (4.5), further implies that

$$E\left\{\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\|\right\} = O\left(\left\{10^{-(\mathrm{SNR}/5)} + 10^{-(\mathrm{SNR}/20)}\right\}^{2^{-(n^2+3n)/2}}\right) \quad (4.10)$$

where we have used (4.7) and (4.2). Therefore, we have shown that the distance between the noise-free and noise-present optimal solution in (3.21) can be bounded and that the bound is a (decreasing) function of the SNR.

Let $\bar{\mathbf{w}}$ be the noise-present optimal equalizer obtained by our new equalization algorithm described in Section III, and let $\bar{\mathbf{w}}_{opt}$ be the optimal noise-free equalizer that is closest to $\bar{\mathbf{w}}$. Now, it remains to bound $\|\bar{\mathbf{w}} - \bar{\mathbf{w}}_{opt}\|$ in terms of $\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\|$. In the absence of noise, $\bar{\mathbf{w}}_{opt}$ lies in the intersection $\mathcal{S} = \mathcal{N}(\bar{\mathbf{G}}_{opt}) \cap \mathcal{H} \cap \mathcal{M}$. Such intersection can be described with a system of equalities:

$$\mathcal{S} = \Big\{\bar{\mathbf{w}}_{opt} \mid \bar{\mathbf{G}}_{opt}\bar{\mathbf{w}}_{opt} = 0,\ \bar{\mathbf{w}}_{opt,j} = \bar{\mathbf{w}}_{opt,k}\bar{\mathbf{w}}_{opt,\ell},\ \bar{\mathbf{w}}_{opt,m} = 1\Big\} \quad (4.11)$$

where $m = 2n + \binom{n}{2} + 1$ is the length of $\bar{\mathbf{w}}_{opt}$, n is the equalizer length, and $\bar{\mathbf{w}}_{opt,j}$ represents the jth component of $\bar{\mathbf{w}}_{opt}$. The first equality, which we will refer to as $h_1(\bar{\mathbf{w}}_{opt}) = 0$, corresponds to the null-space $\mathcal{N}(\bar{\mathbf{G}}_{opt})$; the second set of equalities, which we denote by $h_i(\bar{\mathbf{w}}_{opt}) = 0, 2 \le i \le m - n$, corresponds to the rank-one manifold $\mathcal{M}$; and the last equality, which we write as $h_s(\bar{\mathbf{w}}_{opt}) = 0, s = m - n + 1$, corresponds to the hyperplane $\mathcal{H}$. We need the following lemma.

Lemma IV.2: Consider a set defined by

$$\mathcal{S} = \{\bar{\mathbf{w}}_{opt} \in X \mid \mathbf{h}(\bar{\mathbf{w}}_{opt}) = 0\} \quad (4.12)$$

where $\mathbf{h} = (h_1, h_2, \ldots, h_s)^T$, and each h_i is a real-valued analytic function defined on some open set $X \subseteq \mathbb{R}^m$. Then, for any $\rho > 0$, there exist some constants $\alpha > 0$ and $\beta \in (0, 1]$ such that the following error bound relation holds:

$$\mathrm{dist}(\bar{\mathbf{v}}, \mathcal{S}) \le \alpha\|\mathbf{h}(\bar{\mathbf{v}})\|^{\beta}, \quad \forall\, \bar{\mathbf{v}} \in X,\ \|\bar{\mathbf{v}}\| \le \rho. \quad (4.13)$$

A detailed proof of Lemma IV.2 is given in [26].

Recall $\bar{\mathbf{w}}$ is the noise-present optimal equalizer obtained by our new equalization algorithm described in Section III. Since $\|\bar{\mathbf{w}}\|$ is bounded by a constant ρ, we are in a position to apply Lemma 4.2 to obtain

$$E\left\{\|\bar{\mathbf{w}} - \bar{\mathbf{w}}_{opt}\|\right\} \le \alpha E\left\{\|\mathbf{h}(\bar{\mathbf{w}})\|^{\beta}\right\} \quad (4.14)$$

where $\|\mathbf{h}(\bar{\mathbf{w}})\|^2 = h_1^2(\bar{\mathbf{w}}) + \cdots + h_s^2(\bar{\mathbf{w}})$. Since $\bar{\mathbf{w}}$ lies in the intersection $\mathcal{H} \cap \mathcal{M}$, we have that $h_i(\bar{\mathbf{w}}) = 0$, for $i = 2, \ldots, s$. Moreover

$$\begin{aligned}\|h_1(\bar{\mathbf{w}})\| &= \|\bar{\mathbf{G}}_{opt}\bar{\mathbf{w}}\| = \|(\bar{\mathbf{G}}_{opt} - \mathbf{G}_{opt})\bar{\mathbf{w}} + \mathbf{G}_{opt}\bar{\mathbf{w}}\| \\ &\le \|(\bar{\mathbf{G}}_{opt} - \mathbf{G}_{opt})\bar{\mathbf{w}}\| + \|\mathbf{G}_{opt}\bar{\mathbf{w}}\| \\ &\le \|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\|\|\bar{\mathbf{w}}\| + \|\mathbf{G}_{opt}\bar{\mathbf{w}}\| \\ &= O\left(\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\| + \gamma\|\bar{\mathbf{w}}\|\right) \quad (4.15)\end{aligned}$$

where we used the fact that $\|\bar{\mathbf{w}}\|$ is bounded and that $\bar{\mathbf{w}}$ lies in the subspace spanned by those eigenvectors of $\mathbf{G}_{opt}$ whose eigenvalues are no more than γ. Here, γ denotes the threshold value used by the alternating projection algorithm. Suppose the null space of $\bar{\mathbf{G}}_{opt}$ is denoted by $\mathcal{N}(\bar{\mathbf{G}}_{opt})$. Since $\mathbf{G}_{opt}$ is a perturbed version of $\bar{\mathbf{G}}_{opt}$, $\mathbf{G}_{opt}$ will have a set of small eigenvalues whose corresponding eigenvectors span a "noise" subspace closely approximating $\mathcal{N}(\bar{\mathbf{G}}_{opt})$. Assume that γ is chosen such that the "noise" subspace of $\mathbf{G}_{opt}$ is correctly identified and then that γ is in the order of the small eigenvalues of $\mathbf{G}_{opt}$; we then have $\gamma = O(\|\bar{\mathbf{G}}_{opt} - \mathbf{G}_{opt}\|)$. Combining this with (4.15) yields

$$\begin{aligned}\|\mathbf{h}(\bar{\mathbf{w}})\| &= \|h_1(\bar{\mathbf{w}})\| \\ &= O(\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\| + \gamma\|\bar{\mathbf{w}}\|) = O(\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\|).\end{aligned}$$

As a result, we have from (4.14)

$$\begin{aligned}E&\Big\{\|\bar{\mathbf{w}} - \bar{\mathbf{w}}_{opt}\|\Big\} \\ &\le \alpha E\left\{\|h_1(\bar{\mathbf{w}})\|^{\beta}\right\} \\ &= O\left(E\left\{\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\|^{\beta}\right\}\right) \\ &\le O\left(E\left\{\|\mathbf{G}_{opt} - \bar{\mathbf{G}}_{opt}\|\right\}^{\beta}\right) \\ &= O\left(\left\{10^{-(\mathrm{SNR}/5)} + 10^{-(\mathrm{SNR}/20)}\right\}^{\beta 2^{-(n^2+3n)/2}}\right)\end{aligned}$$

where the third step follows from Jensens inequality for the concave function $x^{\beta}(\beta \in (0,1])$, and the last step follows from (4.10). This shows that the expected distance between the noise-free and noise-present optimal equalizer converges to zero exponentially as the SNR increases.

V. Fractionally Spaced Equalizers

In this section, we briefly present an extension of our convex SDP method for blind equalization of CM signals for the case of fractionally spaced CM equalization. It is well known that *baud spaced* linear equalizers (TSEs) can be inefficient or even ineffective when applied to linear channels with zeros on or near the unit circle and that fractionally spaced linear equalizers (FSEs) are often more appropriate for such scenarios. In particular, it has been shown [16] that constant modulus FSEs (CMA-FSEs) can efficiently and effectively equalize certain channels for which constant modulus TSEs (CMA-TSEs) are ineffective. In fact, under some mild identifiability conditions on the channel, the CMA-FSE has been shown to converge globally to a perfect CM equalizer in the absence of noise, provided that the equalizer is longer than a threshold dependent on the channel length and the oversampling factor [16]. Furthermore, the CMA-FSE has been shown to possess "good" local minima in the presence of noise [2], [9], [10]. However, in Section VI-D, we will present simulation results suggesting that the extension of the SDP formulation to the class of FSEs (developed below) performs even better than the standard iterative CMA-FSEs.

The multichannel representation of the CMA-FSE is shown in Fig. 2. The FSE consists of p subequalizers $\mathbf{w}_i, i = 1, \dots, p$, each of which operates on one of the p subchannels at the baud

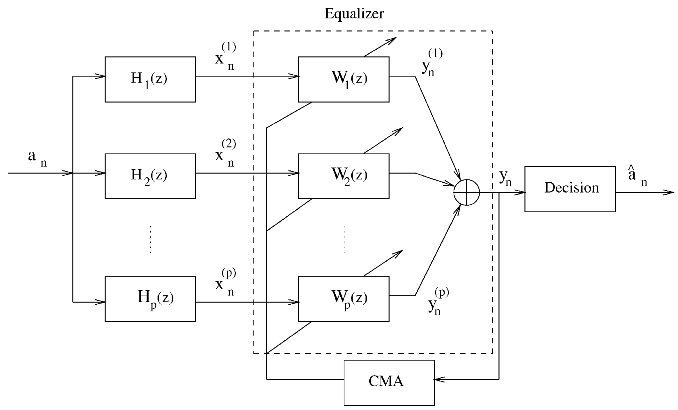


Fig. 2. Multichannel vector representation of the blind adaptive FSE.

rate, $1/T$. The output sequences $x_k^{(i)}$ are given by

$$x_k^{(i)} = \sum_{\ell} a_\ell h_{k-\ell}^{(i)} + w_k^{(i)}, \quad i = 1, \dots, p \tag{5.1}$$

and $h_k^{(i)}$ is a subchannel impulse response. We can express the equalized sequence y_k as

$$y_k = \sum_{i=1}^{p} y_k^{(i)} = \sum_{i=1}^{p} \mathbf{w}_i^H \mathbf{x}_k^{(i)} \tag{5.2}$$

where $\mathbf{x}_k^{(i)}$ is a data snapshot vector. Thus, we can view FSE as a TSE with the actual equalizer being the concatenation of the filters $\mathbf{w}_i$

$$\mathbf{w} = \left(\mathbf{w}_1^H, \dots, \mathbf{w}_p^H\right)^H. \tag{5.3}$$

By defining the combined channel output vector $\mathbf{x}_k$ as

$$\mathbf{x}_k = \left({\mathbf{x}_k^{(1)}}^H, \dots, {\mathbf{x}_k^{(p)}}^H\right)^H \tag{5.4}$$

we can write the equalizer output (5.2) as

$$y_k = \sum_{i=1}^{p} y_k^{(i)} = \mathbf{w}^H \mathbf{x}_k.$$

Substituting (5.3) and (5.4) into (2.3), we can apply the SDP formulation and the post-processing technique presented in Section III to the class of fractionally spaced equalizers.

VI. Implementation and Simulation

We demonstrate the effectiveness of our algorithm through several simulation examples. The value for the threshold γ in the alternating projection algorithm was chosen to be 10^{-7}. We compared our method, which is block based, with the algorithms given in [3]–[5], which are iterative algorithms. From [4], we used an iterative algorithm that minimizes the cost function $J = (1/N)\sum_{k=1}^{N}\big||y_k| - 1\big|$. For our method, we process and decode blocks of N data samples at a time in order to solve for the equalizer, i.e., we need only one block to estimate the equalizer and use it to decode the block. The larger the blocks are, the better the equalizer estimate is. Moreover, by processing the larger data blocks, we can reduce the overall complexity of our algorithm since in that case, the equalizer estimation occurs less frequently. However, if the channel is highly nonstationary, or if the input signal consists of short data packets, one may be forced to work with shorter data blocks. The complexity and computational time of a SDP does not depend on the size of a data block because different block sizes will just give different coefficients $\mathbf{p}_i$ in (3.21). In our simulations, we have used blocks of $N = 500$–1000 samples, although for some applications (e.g., a good quality telephone channel with a moderate-to-high SNR ratio), the block size can be reduced to $N = 150$–200 samples.

We define the intersymbol interference (ISI) as follows:

$$\text{ISI} = \left\| \frac{|\mathbf{t}|}{\max_i |\mathbf{t}_i|} - \mathbf{e} \right\|_2^2$$

where $\mathbf{t}$ is a vector containing the combined channel-equalizer impulse response, $|\mathbf{t}|$ denotes the vector obtained by taking the component wise absolute value of $\mathbf{t}$, $\mathbf{t}_i$ denotes the ith component of the vector $\mathbf{t}$, and $\mathbf{e}$ is a vector with 1 in the position $\arg\max_i |\mathbf{t}_i|$ and zeros elsewhere. For the class of fractionally spaced equalizers, we compared our method with the (iterative) CMA-FSE method of [16] and the (block-based) analytic CMA (ACMA) method presented in [12], [13].

For a length n equalizer, the size of the matrix variable $\mathbf{G}$ in the SDP in (3.21) is $O(n^2) \times O(n^2)$, and in the worst case, solving an SDP of this size requires $O(n^{12.5})$ arithmetic operations. However, the given bound is fully general. In the optimization community, it is generally accepted that these algorithms perform much better in practice, i.e., the given worst case bound is loose. In addition, this estimate does not take into account the fact that the SDP (3.21) is extremely sparse. It is quite conceivable that a specially tailored SDP solver for this problem could be developed. For instance, since the cost function $f(\mathbf{w})$ in (3.1) is a fourth-order polynomial that does not have the third- and the first-order part, it is possible to omit $\mathbf{u}_{(1)}$ in the definition of $\bar{\mathbf{w}}$ in (3.6). This will, in turn, significantly decrease the complexity of our algorithm. In all our examples, the SDPs were solved using a general purpose interior point optimization code SeDuMi [15] developed in Matlab. In our simulations, all the SDPs were solved on a 600-MHz Pentium III PC. The solution of each SDP required a CPU time ranging from a few seconds to a few minutes, depending on the equalizer length and signal constellation. Finally, in all of our simulations, we found that at most five alternating projection iterations were needed in the post-processing step.

A. Good Quality Telephone Channel

In this example, we consider the transmission of BPSK symbols through the channel with impulse response vector

$$\mathbf{h} = [0.04, -0.05, 0.07, -0.21, -0.5 \quad 0.72, 0.36, 0.21, 0.03, 0.07]^T$$

which is a typical response of a good-quality telephone channel [17]. The length of equalizers was chosen to be 11, and the step size parameter for the iterative equalization algorithms was chosen to be 5×10^{-3}, which turned out to give the best results. We allowed 2000 samples for the adaptation of weight coefficients for the methods in [3] and [4], while we processed and decoded blocks of 500 samples at a time for our method. We can see from Fig. 3 that the performance of the iterative methods in [3] and [4] depends on the initialization of the equalizer parameters. The curve denoted by CMA corresponds to the case when we use partial knowledge of the channel impulse response for initialization, i.e., we initialize the equalizer with a single "spike" time-aligned with the center of mass of the channel response. However, if such a knowledge is not available and the spike does not coincide with the channel responses center of mass, the iterative algorithms may degrade in performance, as is shown with the curves CMA1 and CMA3. These curves were generated using equalizers initialized with a spike in the first and third element of the vector, respectively. In all cases, the simulation results indicate that our method achieves better average ISI suppression and a lower bit-error rate. This improved performance is achieved while requiring fewer samples than the algorithms in [3] and [4]. In general, the performance of the adaptive methods in [3] and [4] depends on the step size parameter. For this particular good telephone channel, fine tuning of the step size parameter is not necessary (step size tuning is required for channels with severe ISI), but some approximate channel knowledge is still needed to initialize the equalizer.

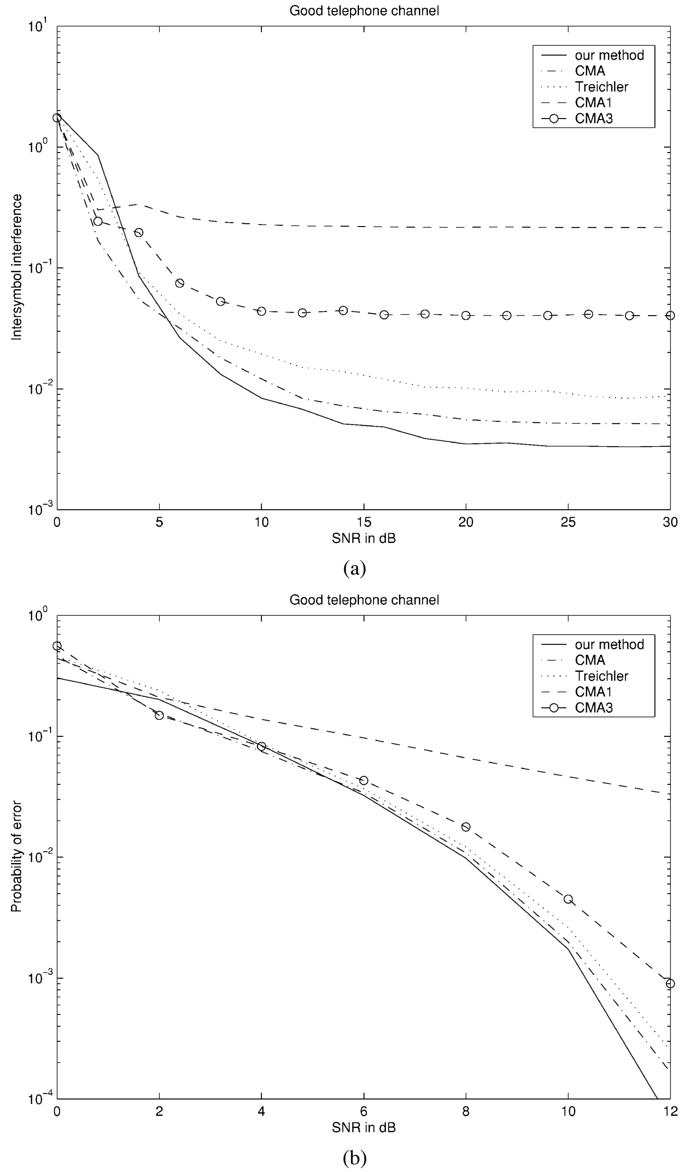


Fig. 3. Intersymbol interference and probability of error versus SNR.

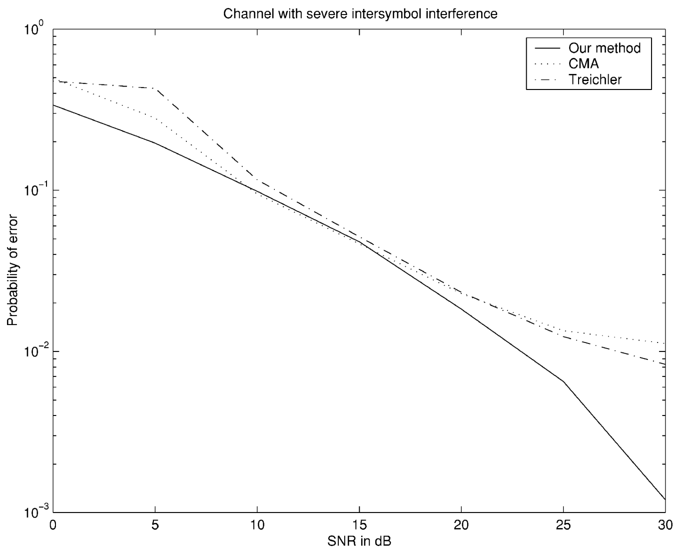


Fig. 4. Probability of error versus SNR for a channel with severe ISI.

B. Channel With Severe Intersymbol Interference

Here, again from [17], we consider the case of a BPSK signal transmitted through a channel with impulse response vector $\mathbf{h} = [0.407, 0.815, 0.407]^T$. This channel corresponds to a channel with severe ISI. This time, we have used 12-tap equalizer for all methods, and we processed 1000 samples at a time for our method. Results shown in Fig. 4 correspond to the case where the iterative equalizers in [3] and [4] are initialized with a single spike time-aligned with the center of mass of the channel response. Again, we see that our method achieves a lower average bit-error rate. If an erroneous initial point is chosen for the iterative methods, the equalizer parameters converge more often to undesirable local minima, which results in poor performance. At the same time, for this particular channel fine-tuning of the step size parameter in the iterative algorithms is necessary to achieve the illustrated performance. In particular, the range of desirable values of the step size parameter is SNR dependent.

C. Non-Minimum Phase Channel

In this example we compare our method with the method proposed in [5]. We consider the case where a communication source transmits a sequence of QPSK symbols through an unknown nonminimum phase channel with impulse response

$$\mathbf{h}_i = \begin{cases} 0, & i < 0 \\ -0.4, & i = 0 \\ 0.84 \times 0.4^{i-1}, & i > 0. \end{cases} \tag{6.1}$$

In our test scenarios, the channel was of length 7. We have used a six-tap equalizer and processed 1000 samples at a time for our method and six- and 12-tap equalizers (denoted by Shalvi6 and Shalvi12, respectively) for the iterative method proposed in [5]. The iterative method was initialized with a single spike in the middle of the equalizer. The value of the step size parameter was set to 7×10^{-4}, which turned out to give the best performance. The final values of the objective function $f(\mathbf{w})$ (averaged over 1000 Monte Carlo runs) for different levels of SNR are shown in Table I.

From Table I, we can see that our method achieves a lower value of the objective function. This leads to better intersymbol interference suppression and better bit-error rate performance (averaged over 1000 Monte Carlo runs), as shown in Table II. [The symbol "$-$" implies that all transmitted bits were recovered successfully during the 1000 Monte Carlo runs.] If we were to initialize the iterative method with a spike set at the beginning of the equalizer ('Shalvi6-ini'), its performance would degrade dramatically. From Table II, we can also see that a 12-tap equalizer performs worse than a six-tap equalizer (denoted by Shalvi6 and Shalvi12, respectively). This is due to the fact that longer equalizers for the iterative method proposed in [5] require more data samples for the initial "training" period.

TABLE I
OBJECTIVE VALUES $f(\mathbf{w}_{opt})$ VERSUS SNR

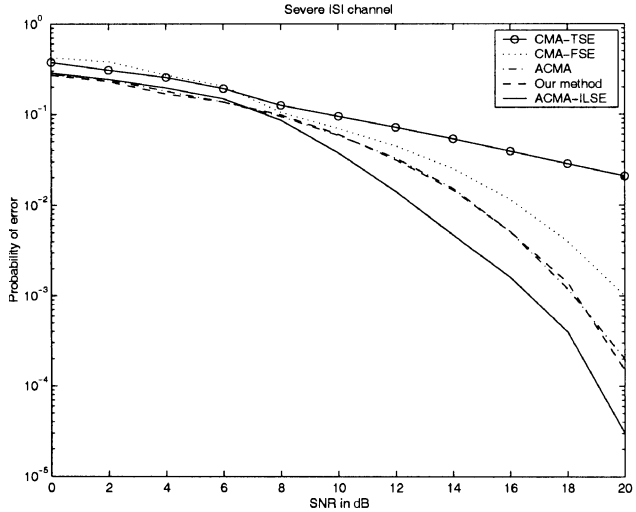
Methods	0dB	4dB	8dB	12dB	16dB	20dB	24 dB
Ours	0.406	0.316	0.198	0.102	0.044	0.019	0.011
Shalvi6	4.331	1.180	0.394	0.153	0.069	0.037	0.025
Shalvi12	4.900	1.199	0.396	0.159	0.068	0.036	0.023
Shalvi6-ini	4.538	1.420	0.628	0.379	0.296	0.271	0.253

TABLE II
BIT-ERROR RATE VERSUS SNR

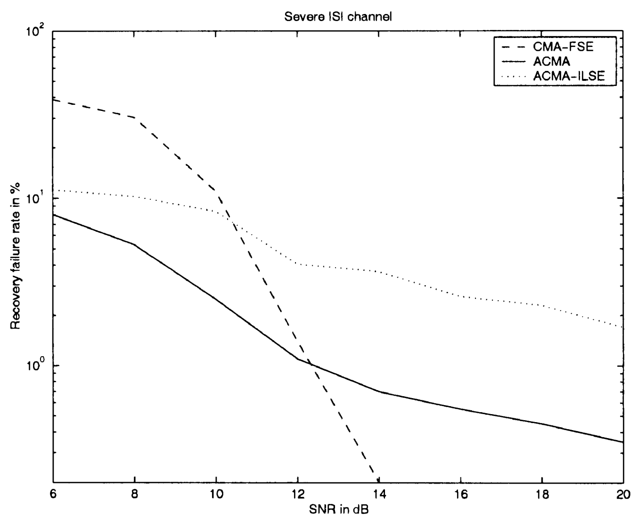
Methods	0dB	4dB	8dB	12dB	16dB	20dB	24 dB
Ours	0.344	0.122	0.014	0.0001	–	–	–
Shalvi6	0.357	0.128	0.018	0.0002	–	–	–
Shalvi12	0.5	0.140	0.020	0.0004	–	–	–
Shalvi6-ini	0.464	0.220	0.090	0.018	0.002	–	–

D. Fractionally Spaced Equalizer

In this section, we compare the performance of several fractionally spaced equalizers with an oversampling factor of $p = 4$. We consider the transmission of BPSK symbols through the baud spaced channel $\mathbf{h} = [0.407, 0.815, 0.407]^T$ that has severe ISI caused by a deep spectral null. In simulation, the fractionally spaced subchannels are obtained by linearly interpolating the baud spaced channel $\mathbf{h}$. For the baud-rate CMA equalizer (CMA-TSE), we used an equalizer of length 12, whereas for the fractionally spaced equalizers, we used shorter equalizers (length 3). For our method and the (block-based) ACMA proposed in [12] and [13], we processed blocks of 1000 samples at a time. For the CMA method [3], the selected step size was 5×10^{-3}. We can see from Fig. 5 that the CMA-FSE outperforms the CMA-TSE, as might be expected [16]. (In this scenario, the CMA-FSE is globally convergent to a perfect CM equalizer in the absence of noise.) However, our SDP based CM equalization method achieves better performance with a lower bit-error rate (BER). The ACMA method proposed in [12] seems to have very similar performance to our method. However, it is important to note that in calculating the BER for ACMA, we have counted only the successful runs for ACMA. The percentage of cases in the 2000 Monte Carlo runs where ACMA failed to equalize the channel is defined as the recovery



(a)



(b)

Fig. 5. Probability of error and recovery failure rate versus SNR.

failure rate. These cases were omitted from the BER statistics but are recorded in the right half of Fig. 5. It is quite natural to use the result of ACMA to initialize the iterative least-squares with enumeration algorithm (ILSE) [18], which is a fixed-point iterative algorithm. If successful, ILSE is a conditional maximum likelihood estimator. However, depending on the initialization, ILSE can converge to a local minimum, and restarts are needed, resulting in a higher recovery failure rate, as shown in the right half of Fig. 5. Once again, the cases of recovery failure have been excluded from the BER statistics of CMA-FSE, ACMA, and ILSE. In contrast, our SDP-based CM equalization method did not fail in any of our 2000 trials; therefore, the corresponding BER curve in Fig. 5 represents the true BER statistics for our method.

VII. Conclusion

In this paper, we have shown that blind equalization of constant modulus signals can be expressed as a convex optimization problem. A semidefinite programming formulation was made possible by performing an algebraic transformation on the direct formulation of the equalization problem. The *average* performance of our method is almost the same as the *optimal* performance of classical CMA equalization methods proposed in [3]–[5]. Here, by optimal performance, we mean that we have used *a priori* knowledge of the channel impulse response envelope to aid the initialization and that we have used the optimally tuned stepsize value for those stochastic descent-based CMA equalization methods. Our simulations suggest that the SDP-based CMA equalization method requires fewer samples, which makes our method useful for applications where convergence times requiring thousands of input samples are undesirable.

We point out that our (block-based) method does incur a higher computational cost than the iterative equalization algorithms proposed in [3]–[5]. In order to solve the semidefinite program, we have used the SeDuMi [15] implementation, which is a polynomial-time primal-dual path-following interior-point algorithm. While this Matlab code is an efficient and robust general purpose SDP solver, it does not explicitly exploit the sparsity in our CMA formulation (3.21). For instance, since the cost function $f(\mathbf{w})$ in (3.1) is a fourth-order polynomial which does not have the third- and the first-order part, it is possible to omit $\mathbf{u}_{(1)}$ in the definition of $\bar{\mathbf{w}}$ in (3.6). This may in turn significantly reduce the computational time and memory requirements. In addition, we expect that some of the recently proposed first-order methods [19]–[21] can be efficiently implemented to exploit the sparsity of the SDPs formulated in this paper. We plan to explore this issue in our future research.

Appendix A

In this section, we provide a brief description of convexity, semidefinite programs and interior-point methods.

Convex set: A set $\mathcal{C}$ is *convex* if the line segment between any two points in $\mathcal{C}$ lies in $\mathcal{C}$. That is, for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$ and any θ with $0 \le \theta \le 1$, we have

$$\theta \mathbf{x}_1 + (1-\theta)\mathbf{x}_2 \in \mathcal{C}.$$

Convex function: A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if $\mathbf{dom} f$ is a convex set and if for all $\mathbf{x}, \mathbf{y} \in \mathbf{dom} f$ and θ with $0 \le \theta \le 1$, we have

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \le \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \qquad (8.1)$$

where $\mathbf{dom} f$ denotes the domain of f. Geometrically, this means that the line segment between $(\mathbf{x}, f(\mathbf{x}))$ and $(\mathbf{y}, f(\mathbf{y}))$ lies "above" the graph of f.

A convex optimization problem is an optimization problem of the form

$$\begin{aligned} &\text{minimize } f_0(\mathbf{x}) \\ &\text{subject to } f_i(\mathbf{x}) \le 0, \quad i = 1, \ldots, m \\ &\qquad\qquad \mathbf{a}_j^T \mathbf{x} = b_j, \quad j = 1, \ldots, p \end{aligned} \qquad (8.2)$$

where $f_0, \ldots, f_m$ are all convex functions, and the equality constraint functions are affine. A fundamental property of a convex programming problem is that any locally optimal point is also globally optimal.

SDPs are an important class of convex optimization problems. These problems have the following form:

$$\begin{aligned}&\text{minimize } \mathbf{c}^T\mathbf{x}\\ &\text{subject to } \mathbf{F}(\mathbf{x}) \succeq \mathbf{0}\end{aligned} \tag{8.3}$$

where the inequality

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + x_1\mathbf{F}_1 + \cdots + x_m\mathbf{F}_m \succeq \mathbf{0} \tag{8.4}$$

is called a *linear matrix inequality* (LMI). The problem data are the vector $\mathbf{c} \in \mathbb{R}^m$ and $m+1$ symmetric matrices $\mathbf{F}_0, \ldots, \mathbf{F}_m \in \mathbb{R}^{n\times n}$ and the variables are $\mathbf{x} = [x_1, x_2, \ldots, x_m]^T$. There are good reasons for studying semidefinite programming. First, positive (or positive definite) constraints arise, either directly or indirectly, in a number of important applications. Second, many convex optimization problems, e.g., linear programming and (convex) quadratically constrained quadratic programming, can be cast as semidefinite programs so that semidefinite programming offers a unified way to study the properties of and derive algorithms for a wide variety of convex optimization problems. Most importantly, however, semidefinite programs can be solved very efficiently using recently developed interior-point methods, which are general concepts for transforming constrained into unconstrained optimization problems.

Interior-point methods solve the original constrained optimization problem as a sequence of smooth unconstrained problems. They replace the constraints by a "barrier" function that is smooth and convex over the feasible set and approaches infinity at the boundaries of the feasible set. There are many barrier functions for the semidefinite cone $\chi = \{\mathbf{x} \mid \mathbf{F}(\mathbf{x}) \succ \mathbf{0}\}$, but the following one enjoys many special properties:

$$\phi(\mathbf{x}) \triangleq \begin{cases}\log\det \mathbf{F}(\mathbf{x})^{-1}, & \text{if } \mathbf{F}(\mathbf{x}) \succeq \mathbf{0}\\ +\infty, & \text{otherwise.}\end{cases} \tag{8.5}$$

The interior point methods based on the *central path* are by far the most useful in theory and the most used in practice. For $t \geq 0$, we define

$$\mathbf{x}^*(t) = \arg\min(t\mathbf{c}^T\mathbf{x} + \phi(\mathbf{x})) \tag{8.6}$$

where we assume that minimizer exists and is unique. This assumption reduces the complexity of the development but does not cause an important loss of generality since most of the theory can be developed without it [22]. The curve described by $\mathbf{x}^*(t)$ is called the *central* path and is defined as the set of points $\mathbf{x}^*(t)$, $t \geq 0$, which we call the *central points*. Points on the central path are characterized by the necessary and sufficient optimality condition $\nabla(t\mathbf{c}^T\mathbf{x} + \phi(\mathbf{x})) = 0$. Given a strictly feasible starting point, we can compute $\mathbf{x}^*(t)$ by solving a smooth, unconstrained minimization problem. At each iteration i, we compute the central point $\mathbf{x}^*(t_i)$ starting from the previously computed central point $\mathbf{x}^*(t_{i-1})$ and then increase t by a factor $\mu > 1$, i.e., $t_{i+1} = \mu t_i$. The parameter t gives a relative weight of the objective and barrier function. As $t \to \infty$, $\mathbf{x}^*(t)$ converges to an optimal value. Several methods for finding initial strictly feasible points exist [22]. Almost all interior-point methods approach the optimal point by following the central path, either literally, by returning to the central path periodically, or by keeping some measure for the deviation from the central path below a certain bound.

APPENDIX B

In this section, we give an explicit estimate of the amount of constraint violation ϵ as a function of the SNR. Recall, we partition the constraint violation vector $\boldsymbol{\epsilon} = \bar{\mathbf{p}} - \mathbf{p}$ as $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_{(4)}^T, \boldsymbol{\epsilon}_{(2)}^T, \boldsymbol{\epsilon}_{(0)}^T)^T$. Obviously, $\boldsymbol{\epsilon}_{(0)} = 0$ since $\mathbf{p}_{(0)} = \bar{\mathbf{p}}_{(0)} = 1$. Since the received data vector $\mathbf{x}_k$ can be written as $\mathbf{x}_k = \mathbf{d}_k + \mathbf{v}_k$, where $\mathbf{d}_k$ is the output of the channel, we can write $\mathbf{p}_{(2)}$ and $\bar{\mathbf{p}}_{(2)}$ as in (9.1), shown at the bottom of the page, where $x_{k,j} = x_{k-j+1}$. Therefore, $\boldsymbol{\epsilon}_{(2)} = \bar{\mathbf{p}}_{(2)} - \mathbf{p}_{(2)}$ will have n terms (where n is the equalizer length) of the form

$$-\frac{2}{N}\sum_{k=1}^{N}(v_{k,i}^2 + 2v_{k,i}d_{k,i}), \quad 1 \leq i \leq n \tag{9.2}$$

and $n(n-1)/2$ terms of the form

$$-\frac{4}{N}\sum_{k=1}^{N}(d_{k,i}v_{k,j} + d_{k,j}v_{k,i} + v_{k,i}v_{k,j}), \quad 1 \leq i < j \leq n. \tag{9.3}$$

Since the sequence $\mathbf{v}_k$ is an additive white Gaussian noise with zero mean and variance σ_v^2, the expected value of $\boldsymbol{\epsilon}_{(2)}$ is equal to

$$\begin{aligned}E\left\{\boldsymbol{\epsilon}_{(2)}\right\} =& E\left\{\mathbf{p}_{(2)} - \bar{\mathbf{p}}_{(2)}\right\}\\ =& (\underbrace{\sigma_v^2, \ldots, \sigma_v^2}_{n\text{ terms}}, \underbrace{0, \ldots, 0}_{\binom{n}{2}\text{ terms}})^T = O(\sigma_v^2).\end{aligned} \tag{9.4}$$

$$\begin{aligned}\mathbf{p}_{(2)} =& -\frac{2}{N}\left(\sum_k d_{k,1}^2, \ldots, \sum_k d_{k,n}^2, 2\sum_k d_{k,1}d_{k,2}, \ldots, 2\sum_k d_{k,n-1}d_{k,n}\right)^T\\ \bar{\mathbf{p}}_{(2)} =& -\frac{2}{N}\left(\underbrace{\sum_k x_{k,1}^2, \ldots, \sum_k x_{k,n}^2}_{n\text{ terms}}, \underbrace{2\sum_k x_{k,1}x_{k,2}, \ldots, 2\sum_k x_{k,n-1}x_{k,n}}_{\binom{n}{2}\text{ terms}}\right)^T\end{aligned} \tag{9.1}$$

In a similar fashion, we can bound $\epsilon_{(4)}$ by partitioning it into five groups:

$$\epsilon_{(4)} = \Big(\underbrace{\epsilon_{(4)1}^T}_{n \text{ terms}}, \underbrace{\epsilon_{(4)2}^T}_{n(n-1)}, \underbrace{\epsilon_{(4)3}^T}_{n(n-1)/2}, \underbrace{\epsilon_{(4)4}^T}_{n\binom{n-1}{2}}, \underbrace{\epsilon_{(4)5}^T}_{\binom{n}{4} \text{ terms}} \Big)^T \quad (9.5)$$

each corresponding to the coefficients of the terms u_i^4, $u_i^3 u_j$, $u_i^2 u_j^2$, $u_i^2 u_j u_l$, $u_i u_j u_l u_m$, and $i \neq j \neq \ell \neq m$, respectively (recall that we are considering the case of a real-valued equalization so that $\mathbf{u} = \mathbf{w}$). The first group $\epsilon_{(4)1}$ consists of n terms of the form

$$\frac{1}{N}\sum_{k=1}^{N}(x_{k,i}^4 - d_{k,i}^4) = \frac{1}{N}\sum_{k=1}^{N} A_{k,i}, \quad i = 1, \ldots, n \quad (9.6)$$

where

$$A_{k,i} = v_{k,i}^4 + 4d_{k,i}^3 v_{k,i} + 4d_{k,i} v_{k,i}^3 + 6d_{k,i}^2 v_{k,i}^2. \quad (9.7)$$

The expected value of (9.7) is

$$\begin{aligned} E\left\{v_{k,i}^4\right\} &+ 4E\left\{d_{k,i}^3\right\} E\left\{v_{k,i}\right\} \\ &+ 4E\left\{d_{k,i}\right\} E\left\{v_{k,i}^3\right\} \\ &+ 6E\left\{d_{k,i}^2\right\} E\left\{v_{k,i}^2\right\}, \quad i = 1, \ldots, n. \end{aligned} \quad (9.8)$$

Using the fact that the v_k are i.i.d Gaussian random variables with zero mean and variance σ_v^2, we have that

$$E\left\{\epsilon_{(4)1}\right\} = O\left(\sigma_v^4 + \sigma_v^2\right). \quad (9.9)$$

The same analysis can be done for the remaining four groups, resulting in

$$E\left\{\epsilon\right\} = O(\sigma_v^4 + \sigma_v^2) \quad (9.10)$$

which implies that the expected value of the amount of constraint violation can be bounded as a function noise power or, equivalently, as a function of the SNR.

Next, we need to estimate $E\left\{\|\epsilon\|\right\}$ since we will bound $\|\bar{\mathbf{G}}_{opt} - \mathbf{G}_{opt}\|$ by $E\left\{\|\epsilon\|^{2^{-d}}\right\}$. This is done in a similar fashion as above. Here, we give an explicit analysis only for the $\epsilon_{(4)1}$ part of the ϵ vector. The same analysis can be done for the remaining parts. According to (9.6), we have the following:

$$\begin{aligned} E\left\{\|\epsilon_{(4)1}\|^2\right\} &= E\left\{\epsilon_{(4)1}^T \epsilon_{(4)1}\right\} \\ &= nE\left\{\left(\frac{1}{N}\sum_{k=1}^{N} A_{k,1}\right)^2\right\}. \end{aligned} \quad (9.11)$$

For the ease of exposition, we denote $A_{k,i}$ by A_i. Then, we can write

$$\begin{aligned} E\left\{\|\epsilon_{(4)1}\|^2\right\} &= \frac{n}{N^2} E\left\{(A_1 + A_2 + \cdots + A_N)^2\right\} \\ &= \frac{n}{N^2} E\Big\{\Big((A_1^2 + A_2^2 + \cdots + A_N^2) \\ &\qquad + 2\sum_{i<j} A_i A_j\Big)\Big\} \\ &= \frac{n}{N^2}\left(NE\left\{A_1^2\right\} + 2\tbinom{N}{2} E\{A_1 A_2\}\right) \\ &= \frac{n}{N} E\left\{A_1^2\right\} + n\frac{N-1}{N} E\{A_1 A_2\} \end{aligned}$$

where $\binom{N}{2} = N(N-1)/2$. According to (9.7), we have that

$$\begin{aligned} E\left\{A_1 A_2\right\} = E\big\{&\left(v_{k,2}^4 + 4v_{k,2}^3 d_{k,2} + 6d_{k,2}^2 v_{k,2}^2 + 4v_{k,2} d_{k,2}^3\right) \\ &\times \left(v_{k,1}^4 + 4v_{k,1}^3 d_{k,1} + 6d_{k,1}^2 v_{k,1}^2 + 4v_{k,1} d_{k,1}^3\right)\big\}. \end{aligned}$$

Using the fact that the v_k are i.i.d Gaussian random variables with zero mean and variance σ_v^2, we have that

$$E\left\{A_1 A_2\right\} = 9\sigma_v^8 + 36\sigma_v^6 + 36\sigma_v^4 = O(\sigma_v^8 + \sigma_v^4).$$

A similar analysis can be done to show that $E\left\{A_1^2\right\} = O(\sigma_v^8 + \sigma_v^2)$, implying

$$E\left\{\|\epsilon_{(4)1}\|^2\right\} = O(\sigma_v^8 + \sigma_v^2).$$

The same estimates can be established for the remaining parts of ϵ. Therefore, assuming σ_v is bounded, we obtain the final result

$$E\left\{\|\epsilon\|\right\} = O(\sigma_v^4 + \sigma_v). \quad (9.12)$$

REFERENCES

[1] H. Vikalo, B. Hassibi, B. Hochwald, and T. Kailath, "Optimal training for frequency-selective fading channels," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, UT, May 2001.

[2] C. R. Johnson Jr, P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas, "Blind equalization using the constant modulus criterion: A review," *Proc. IEEE*, vol. 86, no. 10, pp. 1927–1950, Oct. 1998.

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

[4] J. R. Treichler and M. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP–31, pp. 459–472, Apr. 1983.

[5] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems (channels)," *IEEE Trans. Inform. Theory*, vol. 36, pp. 312–321, Mar. 1990.

[6] J. R. Treichler, V. Wolff, and C. R. Johnson Jr, "Observed misconvergence in the constant modulus adaptive algorithm," in *Proc. 25th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, 1991, pp. 663–667.

[7] Z. Ding, C. R. Johnson Jr, and R. A. Kennedy, "On the (non)existence of local equilibria of Godard blind equalizers," *IEEE Trans. Signal Processing*, vol. 40, pp. 2425–2433, Oct. 1992.

[8] Y. Li, K. J. R. Liu, and Z. Ding, "Length- and cost-dependent local minima of unconstrained blind equalizers," *IEEE Trans. Signal Processing*, vol. 44, pp. 2726–2735, Nov. 1996.

[9] I. Fijalkow, A. Touzni, and J. R. Treichler, "Fractionally spaced equalization using CMA: Robustness to channel noise and lack of disparity," *IEEE Trans. Signal Processing*, vol. 45, pp. 56–66, Jan. 1997.

[10] H. H. Zeng, L. Tong, and C. R. Johnson Jr, "An analysis of constant modulus receivers," *IEEE Trans. Signal Processing*, vol. 47, pp. 2990–2999, Nov. 1999.

[11] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, pp. 49–95, 1996.

[12] A.-J. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 44, pp. 1136–1155, May 1996.

[13] A.-J. van der Veen, "Analytical method for blind binary signal separation," *IEEE Trans. Signal Processing*, vol. 45, pp. 1078–1082, April 1997.

[14] Y. Nesterov, "Squared functional systems and optimization problems," in *High Performance Optimization*, H. Frenk *et al.*, Ed. Boston, MA: Kluwer, 2000, pp. 405–440.

[15] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Meth. Software*, vol. 11–12, pp. 625–653, 1999.

[16] Y. Li and Z. Ding, "Global convergence of fractionally spaced Godard (CMA) adaptive equalizers," *IEEE Trans. Signal Processing*, vol. 44, pp. 818–826, Apr. 1996.

[17] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2000.

[18] S. Talwar, M. Viberg, and A. Paulraj, "Blind separation of synchronous co-channel digital signals using an antenna array—Part I: Algorithms," *IEEE Trans. Signal Processing*, vol. 44, pp. 1184–1197, May 1996.

[19] S. Benson, Y. Ye, and X. Zhang, "Solving large-scale sparse semidefinite programs for combinatorial optimization," Dept. Manage. Sci., Univ. Iowa, Iowa City, IA, 1998.
[20] C. Helmberg and F. Rendl, "A spectral bundle method for semidefinite programming," Konrad-Zuse-Zentrum fuer Informationstechnik Berlin, Berlin, Germany, 1997.
[21] S. Burer, R. D. C. Monteiro, and Y. Zhang, "Interior-point algorithms for semidefinite programming based on a nonlinear programming formulation," Dept. Comput. Applied Math., Rice Univ., Houston, TX, USA, Tech. Rep. TR99-27, 1999.
[22] Z.-Q. Luo, "EE 799: Engineering optimization, Lecture Notes," Dept. Elect. Comput. Eng., McMaster Univ., Hamilton, ON, Canada, 1999.
[23] Y. Li and Z. Ding, "Global convergence of fractionally spaced Godard (CMA) adaptive equalizers," *IEEE Trans. Signal Processing*, vol. 44, pp. 818–826, Apr. 1996.
[24] Z.-Q. Luo and J. Sturm, "Error analysis," in *Handbook of Semidefinite Programming*, H. Wolkowicz, R. Saigal, and L. Vandenberghe, Eds. Dordrecht, The Netherlands: Kluwer, 2000.
[25] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.
[26] Z.-Q. Luo and J. S. Pang, "Error bounds for analytic systems and their applications," *Math. Program.*, vol. 67, pp. 1–28, 1994.
[27] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.

Boris Maričić received the B.Sc. degree in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1999 and the M.Sc. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 2001.

He is currently with the Advanced Signal Processing for Communications Group at McMaster University. His research interests are in communications and signal processing.

Zhi-Quan Luo (SM'03) was born in Nanchang, Jiangxi province, China. He received the B.Sc. degree in applied mathematics in 1984 from Peking University, Beijing, China. From 1984 to 1985, he studied at the Nankai Institute of Mathematics, Tianjin, China. He received the Ph.D. degree in operations research from the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, in 1989.

In 1989, he joined the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, where he is now the Department Chair and holds the Canada Research Chair in Information Processing. His research interests lie in the union of large-scale optimization, information theory and coding, data communications, and signal processing. He is presently serving as an associate editor for *Journal of Optimization Theory and Applications, SIAM Journal on Optimization, Mathematics of Computation, Mathematics of Operations Research, and Optimization and Engineering*.

Prof. Luo is a member of SIAM and MPS. He is an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

Timothy N. Davidson (M'96) received the B.Eng. (Hons. I) degree in electronic engineering from The University of Western Australia (UWA), Perth, in 1991 and the D.Phil. degree in engineering science from the The University of Oxford, Oxford, U.K., in 1995.

He is currently an assistant professor with the Department of Electrical and Computer Engineering McMaster University, Hamilton, ON, Canada. His research interests are in signal processing, communications, and control, with current activity focused on signal processing for digital communication systems. He has held research positions at the Communications Research Laboratory, McMaster University; the Adaptive Signal Processing Laboratory, UWA; and the Australian Telecommunications Research Institute, Curtin University of Technology, Perth.

Dr. Davidson reecived the 1991 J. A. Wood Memorial Prize (for "the most outstanding [UWA] graduand" in the pure and applied sciences) and the 1991 Rhodes Scholarship for Western Australia.