Direct Blind Multiuser Detection for CDMA in Multipath without Channel Estimation

Xiaohua Li and H. Howard Fan, Senior Member, IEEE

Abstract—In this paper, we consider the blind multiuser detection problem for asynchronous DS-CDMA systems operating in a multipath environment. Using only the spreading code of the desired user, we first estimate the column vector subspace of the channel matrix by multiple linear prediction. Then, zero-forcing detectors and MMSE detectors with arbitrary delay can be obtained without explicit channel estimation. This avoids any channel estimation error, and the resulting methods are therefore more robust and more accurate. Corresponding batch algorithms and adaptive algorithms are developed. The new algorithms are extremely near–far resistant. Simulations demonstrate the effectiveness of these methods.

Index Terms—Adaptive equalizers, code division multiaccess, intersymbol interference, multipath channels.

I. INTRODUCTION

BLIND multiuser detection for CDMA system with multipath channels has received much attention and interest recently. It shows great potential for the future wideband CDMA system because of the existence of multipath phenomenon. Blind joint multiuser detection and channel equalization is a good candidate to reduce both multiple access interference (MAI) and intersymbol (or chip)-interference (ISI or ICI) without any training sequences, which will reduce throughput.

Blind method for multiuser detection began with the work of [3]. Assuming no multipath, minimum output energy (MOE) method [3] and subspace method (SS) [4] were presented for blind multiuser detection with the knowledge of only the desired users’ spreading code and (possibly) timing. Some other existing work on joint equalization and multiuser detection methods are based on a priori knowledge of the channel [1], [2]. They will not work properly when ISI cannot be ignored and the multipath channel is not known.

Recently, some approaches for joint blind multiuser detection and blind channel estimation/estimation were presented. The first kind is subspace-based methods [5]–[7], which usually require singular value decomposition (SVD) or eigenvalue decomposition (EVD) of some form of the data correlation matrix, which is computationally costly. Another drawback of the subspace-based approach is that accurate rank determination may be difficult in a practically noisy environment. The second kind is constrained optimization [8], [9], which results in computationally efficient adaptive algorithms. The blind method in [8] is based on minimizing the output energy of a linear filter subject to a constraint to detect the desired user. A major drawback of this approach is that there is a saturation effect in the steady state, which causes a significant performance gap between the converged blind minimum output energy detector and the true MMSE detector [3], [7]. Furthermore, the performance of the algorithm in [8] critically depends on the nonzero magnitude of the selected tap of the channel response. Some improvements are proposed in [9] to find better constraints. There are also some constrained optimization methods based on CMA or Godard’s cost function [18].

Another kind of approach is linear prediction method [10]–[12] or linear prediction like methods [13]–[15]. This approach is promising because linear prediction is computationally efficient and robust. It is shown in [10]–[14] that blind channel identification and multiuser detector estimation can be performed by applying multichannel linear prediction on the transformed channel output data. On the other hand, in [15], least squares smoothing is used first, which is then followed by a subspace method for channel identification. The main idea of the linear prediction-based approach is using the null subspace of the desired user’s spread code matrix to estimate the channel and then to estimate the detector. One possible drawback is that the channel estimation may suffer from system noise and computation errors, which will deteriorate symbol detection.

It is shown in [16] that a direct blind equalizer can be obtained by using linear prediction to estimate the column vector subspace of the channel without estimating the channel itself. In this paper, we will show that the above approach can also be used in the CDMA system with the known spreading code of the desired user. Not explicitly estimating the channel avoids errors in such estimation. The resulting algorithms are therefore more robust and more accurate. Instead of two stages of linear prediction as in [16], we find that only one stage is required in CDMA. Both a zero-forcing (or decorrelating) detector and an MMSE detector can be obtained, resulting in both a batch algorithm and an adaptive algorithm for each detector. A similar idea is also shown in [17], which, however, uses correlation matrix computation followed by constrained optimization. Hence, only batch algorithms are available.

The paper is organized as follows. A baseband CDMA discrete-time model is presented in Section II. Then, in Section III, the linear prediction approach is formulated to accommodate multiple antennas or oversampling beyond the chip rate. Then, new algorithms that do not need channel estimation for both zero-forcing and MMSE are presented in Section IV. Some properties are discussed in Section V. Finally, we will give some simulation experiments in Section VI and conclusions in Section VII.

Manuscript received April 29, 1999; revised September 15, 2000. This work was supported in part by the Army Research Office under Grant DAAD19-00-1-0529. The associate editor coordinating the review of this paper and approving it for publication was Dr. Brian Sadler.

The authors are with the Department of Electrical and Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH 45221-0030 USA (e-mail: h.fan@uc.edu).

Publisher Item Identifier S 1053-587X(01)00075-7.
II. PROBLEM FORMULATION

Consider either a multiple antenna system or a single antenna system with oversampling beyond the chip rate. We allow that the channel spreading may extend over several symbol intervals. Fig. 1 is a block diagram of a discretized CDMA receiver for the $j$th user and the $\ell$th antenna. There are altogether $J$ users and $L$ antennas in the entire system. $b_j(n)$ is the $j$th user’s symbol sequence. $c_j(k)$ is the $j$th user’s spreading code. $g_{j,\ell}(k)$ is the $j$th multipath channel impulse response for the $\ell$th antenna (or the $\ell$th subchannel in case of oversampling), and $v(n)$ is the additive noise. Note that the received signal $x(n)$ has the same form for all users, which includes MAI (with ICI) due to other users and ICI due to the $j$th channel $g_{j,\ell}(k)$ for the $j$th user and the $\ell$th antenna. For blind detection/equalization, $x(n)$ is the only signal that is available. Together with the known $j$th user’s code sequence $c_j(k)$, our objective is to design the receiver for the $j$th user

The chip “modulated” symbol stream for the $j$th user can be expressed as

$$s_j(k) = \sum_n b_j(n) c_j(k - nL_c)$$

where $b_j(n)$ is at the symbol rate $1/T_s$, and $c_j(k)$ and $s_j(k)$ are at the chip rate $1/T_c$, where $L_c$ is the length of the chip sequences. This signal is transmitted through the channel $g_{j,\ell}(k)$, which is assumed to be causal. We consider the asynchronous case. The received signal at antenna $\ell$ is then

$$x_{\ell}(t) = \sum_{j=1}^J \sum_{i} b_j(k) c_j(i - kL_c) g_{j,\ell}(t - iT_c - \tau_{j,\ell})$$

$$+ v(t)$$

where $\tau_{j,\ell}$ is the propagation delay. Assuming we sample at $t = nT_c$ so that the sampled signal has a delay $\delta_{j,\ell}$ (in chips), we have

$$x_{\ell}(n) = \sum_{j=1}^J \sum_{i} b_j(k) c_j(i - kL_c) g_{j,\ell}(n - i - \delta_{j,\ell})$$

$$+ v(n)$$

(2.3)

where the notations $g_{j,\ell}(k) = g_{j,\ell}(kL_c)$ and $x_{\ell}(n) = x_{\ell}(nT_c)$ are used. It is straightforward to show that (2.3) is equivalent to

$$x_{\ell}(n) = \sum_{j=1}^J \sum_{i} b_j(k) h_{j,\ell}^c(n - k - \delta_{j,\ell})$$

$$h_{j,\ell}^c(n) = \sum_{i=0}^{L_c-1} c_j(i) g_{j,\ell}(n - i).$$

(2.5)

Define

$$h_{j,\ell}^c(n) = h_{j,\ell}^c(nL_c + L_c - i - 1), \quad i = 0, 1, \ldots, L_c - 1$$

$$x_{\ell}^c(n) = x_{\ell}(nL_c + L_c - i - 1), \quad i = 0, 1, \ldots, L_c - 1$$

Then, (2.4) can be written in a multichannel fashion as

$$x_{\ell}^c(n) = \sum_{j=1}^J \sum_{i} b_j(k) h_{j,\ell}^c(n - k - i - \delta_{j,\ell})$$

$$+ v_{\ell}(n)$$

(2.8)

where the subscript $i + \delta_{j,\ell}$ is in the modular $L_c$ sense. Stack up $x_{\ell}^c(n)$ for $i = 0, 1, \ldots, L_c - 1$ to obtain

$$\begin{bmatrix} x_{\ell}^c(0) \\ x_{\ell}^c(1) \\ \vdots \\ x_{\ell}^c(L_c-1) \end{bmatrix} = \sum_{j=1}^J \sum_{i} b_j(k) \begin{bmatrix} h_{j,\ell}^c(n - k - i - \delta_{j,\ell}) \\ h_{j,\ell}^c(n - k - i - L_c - 1 - \delta_{j,\ell}) \\ \vdots \\ h_{j,\ell}^c(n - k - i - \delta_{j,\ell} + L_c - 1) \end{bmatrix}.$$

(2.9)

With obvious vector notations, (2.9) can be written as

$$\mathbf{x}_\ell^c(n) = \sum_{j=1}^J \sum_{k} b_j(k) \mathbf{h}_{j,\ell}^c(n - k)$$

$$= \sum_{j=1}^J \sum_{i=0}^{L_c-1} b_j(n - kL_c) \mathbf{h}_{j,\ell}^c(k)$$

(2.10)

where $L_h$ is related to the length of $g_{j,\ell}(k)$ and the delay $\delta_{j,\ell}$. Let the maximum length of $g_{j,\ell}$, $\ell\in\ell$, $j\in\ell$ be $L_h$ (some of $g_{j,\ell}$ may be zero padded). Then, we can choose

$$L_h = \max_{\ell=1}^{\ell} \left[ \frac{L_c + L_g - 1 + \delta_{j,\ell}}{L_c} \right].$$

(2.11)

In fact, $L_h$ can be over estimated in our algorithms.
Now, let
\[ \mathbf{x}(n) = \begin{bmatrix} \mathbf{x}^0(n) \\ \vdots \\ \mathbf{x}^{L-1}(n) \end{bmatrix}. \quad (2.12) \]

Furthermore, stacking up \( \mathbf{x}(n), \mathbf{x}(n-1), \ldots, \mathbf{x}(n-N+1) \) and using (2.10), we have
\[ \mathbf{X}(n) = \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{x}(n-1) \\ \vdots \\ \mathbf{x}(n-N+1) \end{bmatrix} = \sum_{j=1}^{J} \mathcal{H}_j \mathbf{b}_j(n) + \nu(n) \quad (2.13) \]

where we have
\[ \mathcal{H}_j = \begin{bmatrix} \mathbf{h}_j(0) & \cdots & \mathbf{h}_j(L_h-1) & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{h}_j(0) & \cdots & \mathbf{h}_j(L_h-1) \end{bmatrix} \]
\[ \mathbf{b}_j(n) = \begin{bmatrix} \mathbf{b}_j(0) \\ \mathbf{b}_j(1) \\ \vdots \\ \mathbf{b}_j(n-L_h-N+2) \end{bmatrix} \quad (2.14) \]

where \( N \) is related to the length of \( \mathbf{f}_j(k) \), and
\[ \mathbf{h}_j(k) = \begin{bmatrix} \mathbf{h}_{j,\mathbf{f}_j}^0(k) \\ \vdots \\ \mathbf{h}_{j,\mathbf{f}_j}^{L-1}(k) \end{bmatrix}, \quad k = 0, \ldots, L_h - 1. \quad (2.15) \]

Ignoring the noise \( \nu(n) \) for the time being, the received signal vector \( \mathbf{X}(n) = [\mathbf{x}(nL_c + L_c - 1), \mathbf{x}(nL_c + L_c - 2), \ldots, \mathbf{x}(nL_c), \mathbf{x}(nL_c - 1), \ldots, \mathbf{x}(nL_c - (N-1)L_c - 1), \ldots, \mathbf{x}(nL_c - (N-1)L_c)]^T \) can then be expressed, using (2.13), as
\[ \mathcal{X}(n) = [\mathcal{H}_1 \cdots \mathcal{H}_J] \begin{bmatrix} \mathbf{b}_1(n) \\ \vdots \\ \mathbf{b}_J(n) \end{bmatrix}. \quad (2.16) \]

Define
\[ \mathcal{H} = [\mathcal{H}_1 \cdots \mathcal{H}_J]. \quad (2.17) \]

\( \mathcal{H} \) is of dimension \( NLL_c \times J(L_h + N - 1) \). Similar to other existing multiuser detection approaches [6]–[14] full column rank of \( \mathcal{H} \) is assumed, for which a necessary condition is choosing \( N \) such that
\[ NLL_c \geq J(L_h + N - 1). \quad (2.18) \]

A more detailed discussion on the conditions for full column rank of \( \mathcal{H} \) would be similar to those of [6]–[14] and is omitted here.

Now, we need to express matrix \( \mathcal{H}_j \) in terms of the spreading code \( \mathbf{c}_j(k) \) and the channel impulse response \( \mathbf{g}_j(k) \). The actual channel coefficients are from \( \mathbf{g}_j(0) \) to \( \mathbf{g}_j(L_g - 1) \), where \( L_g \) is the maximum length of all \( J \) channels. Note that \( L_g \) may be longer than one symbol duration, i.e., the channel length may be larger than the spreading ratio \( L_e [14] \). Then, from (2.5)
\[ \begin{bmatrix} h_j^e(L_e + L_g - 2) \\ \vdots \\ h_j^e(0) \\ c_j(L_e - 1) \\ \vdots \\ c_j(0) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_j^e(L_g - 1) \\ \vdots \\ \mathbf{g}_j^e(0) \end{bmatrix} \]
\[ = \mathbf{C}_j^e \mathbf{g}_j^e. \quad (2.19) \]

Therefore, we have
\[ \mathbf{h}_j^e(k) = \begin{bmatrix} h_{j,\mathbf{f}_j}^e(k) \\ \vdots \\ h_{j,\mathbf{f}_j}^{L_e + L_c - d_f^e - 1}(k) \end{bmatrix} \]
\[ = \begin{bmatrix} h_{j,\mathbf{f}_j}^e(kL_e + L_c - d_f^e - 1) \\ \vdots \\ h_{j,\mathbf{f}_j}^e(kL_e - d_f^e) \end{bmatrix} \]
\[ = \mathbf{C}_j^e \mathbf{g}_j^e. \quad (2.20) \]

where in the last of the above equations, we use MATLAB representation to denote \( \mathbf{C}_j^e(k) \) as the submatrix of \( \mathbf{C}_j^e \) from row \( kL_e - d_f^e + 1 \) to \( kL_c + L_e - d_f^e \). Hence
\[ \mathbf{h}_j(k) = \begin{bmatrix} \mathbf{h}_{j,\mathbf{f}_j}^e(k) \\ \vdots \\ \mathbf{h}_{j,\mathbf{f}_j}^{L_e - 1}(k) \end{bmatrix} \]
\[ = \begin{bmatrix} \mathbf{C}_j^e(k) \\ \vdots \\ \mathbf{C}_j^e(L_e - 1)(k) \end{bmatrix} \begin{bmatrix} \mathbf{g}_j^0 \\ \vdots \\ \mathbf{g}_j^{L_e - 1} \end{bmatrix} \]
\[ = \mathbf{C}_j(k) \mathbf{g}_j. \quad (2.21) \]

The matrix \( \mathcal{H}_j \) then can be written as
\[ \mathcal{H}_j = \begin{bmatrix} \mathbf{C}_j(0) & \cdots & \mathbf{C}_j(L_h - 1) & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{C}_j(0) & \cdots & \mathbf{C}_j(L_h - 1) \\ \mathbf{g}_j & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{g}_j \end{bmatrix}. \quad (2.22) \]
III. CHANNEL VECTOR SPACE SEPARATION BY LINEAR PREDICTION

In this section, we introduce the linear transformation and linear prediction with the above CDMA system model in multiple antenna situation to set stage for the development of the new algorithms in Section IV. Including L antennas or subsampling beyond the chip rate by L times increases the number of potential users by L times. Without loss of generality, we assume user 1 is the desired user. The basic idea is to extract the information of \( b_1(n - d_f) \) from the mixed signal on which either zero-forcing or MMSE detectors can be estimated. We accomplish this by applying first linear transformation and then linear prediction upon the data vector \( \mathcal{X}(n) \) but without channel identification as in [10]–[15]. Channel equalization can then be performed by some optimization methods. We now work toward this objective.

A. Linear Transformation

Consider the asynchronous CDMA system (2.12)–(2.20). We assume that at first, the timing of the desired user \( j = 1 \) is known through timing recovery (blind timing estimation will be discussed in Section V-B) so that \( d_1^f = 0, \forall f \). In addition, \( L_g \) can be chosen as a conservative upper limit of all possible cases. The equalizer delay \( d_f \) is between 0 and \( L_h + N - 1 \). Since choosing middle value usually has better performance, we can, for example, let \( d_f^t = [(L_h + N)/2] \). For simplicity, we consider only \( d_f \leq N \) case. The \( d_f > N \) case can be similarly obtained.

The channel matrix \( \mathcal{H}_1 \) of the desired user

\[
\mathcal{H}_1 = \begin{bmatrix}
\mathbf{h}_1(0) & \cdots & \mathbf{h}_1(L_h - 1) \\
\vdots & \ddots & \vdots \\
\mathbf{h}_1(0) & \cdots & \mathbf{h}_1(L_h - 1)
\end{bmatrix}
\]  

(3.1)

has \( N \) block rows and \( N + L_h - 1 \) columns. The \( (d_f + 1) \)th column contains \( \mathbf{h}_1(d_f), \cdots, \mathbf{h}_1(0) \). We construct data vector

\[
\mathcal{Y}_1(n) = \begin{bmatrix}
\mathbf{x}(n - d_f + L_h - 1) \\
\vdots \\
\mathbf{x}(n - d_f)
\end{bmatrix}.
\]  

(3.2)

Proposition 1: There exists a full row rank \( LL_c \times L_hLL_c \) matrix \( \mathbf{T} \) such that \( \mathbf{T}\mathcal{Y}_1(n) \) does not contain \( b_1(n - d_f) \).

Proof: From (2.14) and (3.2), in the channel matrix \( \mathcal{H}_1 \), the column vector corresponding to \( b_1(n - d_f) \) in \( \mathcal{Y}_1(n) \) is

\[
\begin{bmatrix}
\mathbf{h}_1(L_h - 1) \\
\vdots \\
\mathbf{h}_1(0)
\end{bmatrix}
= \begin{bmatrix}
\mathbf{C}_1(L_h - 1)\mathbf{g}_1 \\
\vdots \\
\mathbf{C}_1(0)\mathbf{g}_1
\end{bmatrix} = \mathbf{C}_1\mathbf{g}_1.
\]  

(3.3)

The matrix \( \mathbf{C}_1 \) has dimension \( L_hLL_c \times LL_c \). From (2.11), we have \( L_hLL_c - LL_g > LL_c \). Therefore, by choosing \( \mathbf{T} \) as the left singular vectors of \( \mathbf{C}_1 \) corresponding to its zero singular values, we get

\[
\mathbf{T}
\begin{bmatrix}
\mathbf{h}_1(L_h - 1) \\
\vdots \\
\mathbf{h}_1(0)
\end{bmatrix}
= \mathbf{T}\mathbf{C}_1\mathbf{g}_1 = 0.
\]

Hence, the proposition is proved.

Since \( \mathbf{T} \) is obtained from the known spreading code matrix of the desired user, it can be computed off line. Furthermore, \( \mathbf{T}\mathcal{Y}_1(n) \) does not completely cancel other symbol components because different code matrices do not have identical null space.

Using the transformed data vector \( \mathbf{T}\mathcal{Y}_1(n) \), we construct a new data vector

\[
\mathcal{Y}_2(n) = \begin{bmatrix}
\mathbf{x}(n - d_f + L_h - 1 + M_1) \\
\vdots \\
\mathbf{x}(n - d_f + L_h - 1 + 1) \\
\mathbf{T}\mathcal{Y}_1(n) \\
\mathbf{x}(n - d_f - 1) \\
\vdots \\
\mathbf{x}(n - d_f - M_2)
\end{bmatrix}
\]  

(3.4)

where \( M_1 \) and \( M_2 \) satisfy

\[
M_1LL_c \geq J(M_1 + L_h - 1), \quad M_2LL_c \geq J(M_2 + L_h - 1).
\]  

(3.5)

The channel matrix corresponding to \( \mathcal{Y}_2(n) \) has the following structure:

\[
\begin{bmatrix}
\mathcal{H}_{M1} \\
\mathcal{H}_{T1} \\
0 \\
\mathcal{H}_{T2} \\
\mathcal{H}_{M2}
\end{bmatrix}
\]

where \( \mathcal{H}_{M1}, \mathcal{H}_{T1}, \mathcal{H}_{T2}, \mathcal{H}_{M2} \) are channel matrices corresponding, respectively, to \( \mathbf{X}^H(n - d_f + L_h - 1 + M_1) \cdots \mathbf{X}^H(n - d_f + L_h) \) and \( \mathbf{X}^H((n - d_f - 1)) \cdots \mathbf{X}^H(n - d_f - M_2) \) and, hence, with dimensions \( M_1LL_c \times J(M_1 + L_h - 1) \) and \( M_2LL_c \times J(M_2 + L_h - 1) \). The matrix \( [\mathcal{H}_{T1}, 0, \mathcal{H}_{T2}] \) is the channel matrix corresponding to \( \mathbf{T}\mathcal{Y}_1(n) \), with 0 being a zero column vector. The assumption that \( \mathcal{H} \) in (2.17) is full column rank guarantees that \( \mathcal{H}_{M1} \) and \( \mathcal{H}_{M2} \) are each full column rank under (3.5). Therefore, we find the channel matrix of \( \mathcal{Y}_2(n) \) without considering the all-zero column is also full column rank. Note that this assumption is critical to the following linear prediction step.

B. Linear Prediction

We would like to extract the \( b_1(n - d_f) \) part of \( \mathcal{X}(n) \) using linear prediction. Considering that \( \mathcal{X}(n) \) contains \( b_1(n - d_f) \), whereas \( \mathcal{Y}_2(n) \) does not, we define the following linear prediction problem:

\[
\mathbf{e}_2(n) = \mathcal{X}(n) - \mathbf{P}\mathcal{Y}_2(n)
\]  

(3.6)

where \( \mathbf{P} \) has dimension \( NLL_c \times MLL_c \) with \( M = M_1 + M_2 + 1 \). Assume that the symbols \( b_j(n) \) are uncorrelated in time and that \( b_1(n), \cdots, b_f(n) \) are mutually uncorrelated with variances (powers) \( A_1, \cdots, A_f \). We define

\[
\mathcal{X}(n) \overset{\Delta}{=} \mathcal{H}_1b_1(n - d_f) + \mathbf{H}_1\mathbf{b}_2(n)
\]  

(3.7)

where \( \mathbf{b}_2(n) \) contains all symbol components in \( [b_2^H(n) \cdots b_f^H(n)]^H \) [see (2.16)] except for \( b_1(n - d_f) \). \( \mathbf{H}_1 \) is the column vector of the channel matrix \( \mathcal{H} \) corresponding to \( b_1(n - d_f) \), whereas all other columns of \( \mathcal{H} \) (2.16) comprise \( \mathbf{H}_1 \). We have the following linear prediction results.

Proposition 2: The optimal linear prediction matrix \( \mathbf{P} \) gives

\[
\mathbf{e}_1(n) = \mathbf{H}_1b_1(n - d_f).
\]  

(3.8)
Proof: Let $\tilde{H}_2$ and $\tilde{b}_2(n)$ denote the channel matrix and symbol vector corresponding to $Y_2(n)$ in (3.4), i.e., $Y_2(n) = \tilde{H}_2 \tilde{b}_2(n)$, where $b_2(n)$ does not contain $b_2(n-d_f)$. From (3.6) and (3.7), we have

$$\varepsilon_1(n) = H_1b_1(n-d_f) + ([0 \quad \tilde{H}_1 \quad 0] - P\tilde{H}_2)\tilde{b}_2(n) \quad (3.9)$$

where the $0$ matrices are with proper dimensions due to the fact that $b_1(n) \notin \tilde{b}_2(n)$. Then

$$E[\varepsilon_1(n)\varepsilon_1^H(n)] = H_1A_1H_1^H + ([0 \quad \tilde{H}_1 \quad 0] - P\tilde{H}_2)\cdot\text{diag}\{A_i\}([0 \quad \tilde{H}_1 \quad 0] - P\tilde{H}_2)^H.$$}

Minimizing $E[\varepsilon_1(n)\varepsilon_1^H(n)]$ over $P$ then gives

$$\tilde{H}_2\text{diag}\{A_i\}([0 \quad \tilde{H}_1 \quad 0] - P\tilde{H}_2)^H = 0.$$}

Because $\tilde{H}_2$ is of full column rank, we have

$$([0 \quad \tilde{H}_1 \quad 0] - P\tilde{H}_2) = 0. \quad (3.10)$$

Hence, from (3.9) and (3.10), one easily obtains (3.8).

Let

$$\varepsilon_2(n) = \mathcal{X}(n) - \varepsilon_1(n). \quad (3.11)$$

Then, from (3.7) and (3.8), we have

$$\varepsilon_2(n) = \tilde{H}_1\tilde{b}_1(n). \quad (3.12)$$

Hence, the channel matrix vector space is separated into two subspaces by linear prediction.

The solution of $P$ and linear prediction errors can also be explicitly represented by the data correlations. Rewrite the linear prediction problem in (3.6) as

$$\varepsilon_1(n) = [1 \quad -P] \begin{bmatrix} \mathcal{X}(n) \\ Y_2(n) \end{bmatrix} \quad (3.13)$$

and let

$$R = E \begin{bmatrix} \mathcal{X}(n) \\ Y_2(n) \end{bmatrix} \begin{bmatrix} \mathcal{X}(n)^H \\ Y_2(n)^H \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \quad (3.14)$$

where $R_{11}$ and $R_{22}$ are with dimensions $NLL_c \times NLL_c$ and $MNL_c \times MNL_c$, respectively. Then, it is well known that the optimal solution for the linear prediction problem (3.13) is [16]

$$P = R_{12}R_{22}^+ \quad (3.15)$$

$$E[\varepsilon_1(n)\varepsilon_1^H(n)] = A_1H_1H_1^H = R_{11} - R_{12}R_{22}^+R_{21} \quad (3.16)$$

where $(\cdot)^+$ denotes pseudoinverse. Note that the solution for $P$ is not uniquely defined, and we use the minimum norm solution in (3.15). After some straightforward deduction [16], we also have

$$E[\varepsilon_2(n)\varepsilon_2^H(n)] = \tilde{H}_1\text{diag}\{A_i\}\tilde{H}_1^H = R_{12}R_{22}^+R_{21}. \quad (3.17)$$

IV. Blind Equalization and Multiuser Detection

Based on the above separated channel matrix vector spaces, we can further derive decorrelating and MMSE detectors without explicit channel estimation. Channel estimation and symbol detection are obtained according to (3.8) in [10]-[14]. However, a robust solution requires correlation estimation and singular vector estimation. A simplified approach is used in [10]-[12], which may introduce larger errors in channel estimation and symbol detection. Theoretically, the robust solution based on (3.8) by channel estimation requires high SNR and long data sequences. Therefore, direct detector estimation by some optimization methods without explicit channel estimation [based on (3.8)] may be more reliable. In this section, we develop zero-forcing as well as MMSE detectors based on this line of thinking.

A. Blind Zero-Forcing Detection/Equalization

From (3.7) and the definition of zero-forcing equalizer [16], a zero-forcing detector $f$ with dimension $NLL_c \times 1$ satisfies

$$f^H\mathcal{X}(n) = f^H[1 \quad 0]^H.$$

Therefore, we need

$$f^H[1 \quad \tilde{H}_1] = [1 \quad 0]^H. \quad (4.1)$$

Note that for (4.2) to have an exact solution, it is necessary that $[1 \quad \tilde{H}_1] = H$ has more rows than columns, i.e., (2.18) should be satisfied in choosing the detector length.

According to (3.8), (3.16), and (3.17), we find [16] that

$$f^H\tilde{H}_1 \neq 0 \iff f^H[1 \quad \tilde{H}_1] = [1 \quad 0]^H$$

and

$$f^H\tilde{H}_1 = 0 \iff f^H[1 \quad \tilde{H}_1] = [1 \quad 0]^H.$$

Theoretically, $R_{11} - R_{12}R_{22}^+R_{21}$ is with rank 1 due to (3.16). Let $u$ be its left singular vector corresponding to its nonzero singular value; then, $u$ is an estimate of $H_1$ and $f^H\tilde{H}_1 \neq 0$ if $f^H\tilde{H}_1 \neq 0$.

Therefore, to estimate the zero-forcing detector from (4.2), we define

$$B = R_{12}R_{22}^+ \quad (4.5)$$

Then, from (4.2)-(4.4), the zero-forcing detector $f$ satisfies

$$f^H[1 \quad 0] = [1 \quad 0]. \quad (4.6)$$

Hence

$$f^H = [1 \quad 0]B^+. \quad (4.7)$$

Alternatively, we can find the zero-forcing detector based on (4.5) and (4.6) by solving the following equality constrained least squares problem:

$$\min_{f^H} \|f^H[1 \quad 0][R_{12}R_{22}^+R_{21}]\|^2. \quad (4.8)$$
The standard algorithm for solving (4.8) can be found in [21, pp. 585].

The above development is under the noiseless assumption. In the noise case, we need to modify the correlation matrix estimations as

$$\hat{R} = R - \sigma_n^2 I. \quad (4.9)$$

In case when the noise power is not known, an SVD of $R$ can be used to find the signal subspace and noise subspace to estimate the noise power. Similarly, a standard SVD can be used to find $R_e^+$ that is needed in (3.15). For rank estimation in calculating $R_e^+$, one can use, for example, the AIC or the MDL methods of [24]. Note that the adaptive algorithm in the sequel eliminates these steps.

The batch algorithm for zero-forcing detector estimation is outlined below.

- **Algorithm 1: Batch Zero-Forcing Algorithm (BZF)**
  1. From the channel output, compute the correlation matrix $R$ (3.14).
  2. Estimate $\eta$ from (3.16).
  3. Compute zero-forcing detector by (4.5)-(4.7) or by (4.8).

In order to track time-varying channels, we can develop an adaptive algorithm to estimate the blind zero-forcing equalizer/detector. From (3.8) and (3.12), we can solve the following minimization problem to find $f$ that satisfies (4.2)

$$\min_f J(n) = \|f^H \eta(n)\|^2 \quad \text{subject to} \quad \eta^H \eta(n) \neq 0. \quad (4.10)$$

A method for implementing the above constrained optimization problem by unconstrained optimization is

$$\min_f J(n) = \|f^H \eta(n)\|^2 + \alpha(\eta^H \eta(n))^2 \quad (4.11)$$

where $\alpha$ is a weighting factor. Note that the second term is similar to Godard’s cost function [20], and hence, higher than second-order statistics is used. We prefer to find a cost function with second-order statistics so that convergence can be assured. From (3.16), $u$ is the left singular vector of $R_e = E\{\eta(n)\eta^H(n)\}$ corresponding to the nonzero singularvalue. $f$ can therefore be estimated by optimizing

$$\min_f J(n) = \|f^H \eta(n)\|^2 \quad \text{st.} \quad f^H \eta = 1. \quad (4.12)$$

For this constrained adaptive optimization, a well-known method is the Frost algorithm in the array signal processing literature [23]. The adaptation of $f$ by the Frost algorithm is

$$f(n+1) = u(n)u^H(n)^{-1} + [1 - u(n)u^H(n)^{-1}u^H(n)] \cdot f(n) - \mu e_2(n)u^H(n)f(n) \quad (4.13)$$

where $u$ can be estimated adaptively as the column of $R_e(n)$ with the largest norm and $R_e(n+1) = (1 - \lambda)R_e(n) + \lambda e_2^H(n)e_2^H(n)f(n)$ with some forgetting factor $\lambda$. The adaptive algorithm for zero-forcing detector estimation is outlined in the following.

- **Algorithm 2: Adaptive Zero-Forcing Algorithm (AZF)**
  1. Compute the adaptive multichannel linear prediction $\eta_1(n)$ (3.6).
  2. Compute the $\eta_2(n)$ (3.11).
  3. Estimate the max-normed vector $u(n)$ of the correlation matrix (3.16).
  4. Adaptively estimate $f(n)$ (4.13).

Note that in the adaptive algorithm, we do not need to try to reduce noise explicitly. Instead, we use the channel output data vector directly in the linear prediction.

B. Blind MMSE Equalization and Multiuser Detection

Zero-forcing detectors are usually obtained under the noiseless assumption. In reality, they may occasionally magnify noise while performing equalization. When noise is large, the MMSE equalizer/detector may be more desirable. Therefore, we will develop MMSE equalizers/detectors in this subsection. Suppose $X(n)$ contains noise, and the magnitude of the desired symbol $A_1 = 1$. The MMSE detector $m$ with delay $d_f$ and dimension $NLL_e \times 1$ satisfies

$$\min_m \|b_1(n - d_f) - m^H X(n)\|^2. \quad (4.14)$$

After some deduction, one obtains [7]

$$m = R_{11}^{-1} H_1 \quad (4.15)$$

where the matrices are defined as in (3.7) and (3.14). The estimation of $H_1$, or $u$ from (3.16), can be used to obtain

$$m = R_{11}^{-1} u. \quad (4.16)$$

To avoid the direct computation of (4.16), we define a constrained optimization problem similar to (4.12):

$$\min_m \|b_1^H m - m^H R_{11} m\|^2 \quad \text{subject to} \quad m^H u = 1. \quad (4.17)$$

Then, using Lagrange optimization method, one can readily obtain [23]

$$m = (u^H R_{11}^{-1} u)^{-1} R_{11}^{-1} u. \quad (4.18)$$

Comparing (4.18) with the MMSE equalizer (4.16), we find that the only difference is a scalar factor. Thus, the optimization of (4.17) yields an MMSE detector.

The evaluation of (4.17) is similar to the equality constrained least squares problem (4.8). For a normalized $u$, we construct a unitary matrix $Q = [u, u_1, \ldots, u_{NLL_e}]$, e.g., by the QR decomposition of $u$. Then, $m^H Q = [1, g]^T$. The constrained optimization problem becomes the unconstrained problem

$$\min_g [1, g]^T Q^H R_{11} Q [1, g] \quad (4.19)$$

which is similar in form to the linear prediction problem (3.13). The solution is also similar to (3.15). Hence, a batch algorithm is outlined below for blind MMSE detector estimation based on the predictive channel vector space separation.
**Algorithm 3: Batch MMSE Detection (BMMSE)**

1. From the channel output, compute the correlation matrix $\mathbf{R}$ (3.14).
2. Estimate the vector $\mathbf{u}$ from (3.16).
3. Compute MMSE detector $\mathbf{m}$ by (4.17) or (4.19).

In order to adaptively evaluate (4.17), we use the instantaneous value of $\mathbf{R}_{11}$ to obtain

$$\min_{\mathbf{m}} J(n) = ||\mathbf{m}^H \mathbf{X}(n)||^2, \quad \text{subject to} \quad \mathbf{m}^H \mathbf{u} = 1. \quad (4.20)$$

Compared with the zero-forcing detector estimation introduced in Section IV-A, the only difference is that we use $\mathbf{X}(n)$ here instead of $\mathbf{e}(n)$. Therefore, the adaptive method discussed in Section IV-A applies here. To summarize, the adaptation of $\mathbf{m}$ by the Frost algorithm is

$$\mathbf{m}(n+1) = \mathbf{u}(n)^H \mathbf{u}^{-1} + [\mathbf{I} - \mathbf{u}(n)^H \mathbf{u}^{-1} \mathbf{u}(n)^H] \cdot [\mathbf{m}(n) - \mu \mathbf{X}(n) \mathbf{X}^H(n) \mathbf{m}(n)]. \quad (4.21)$$

Therefore, we get the following adaptive algorithm for estimating MMSE detector.

**Algorithm 4: Adaptive MMSE Detection (AMMSE)**

1. Compute the adaptive multichannel linear prediction $\mathbf{e}_1(n)$ (3.6).
2. Estimate the max-normed vector $\mathbf{u}(n)$ from (3.16).
3. Adaptively estimate $\mathbf{m}(n)$ (4.21).

The noise case can be dealt with similarly as Algorithms 1 and 2.

**C. Comparison with Other Methods**

It is interesting to compare our new algorithms with some similar methods. First, we compare our zero-forcing algorithms in Section IV-A with the channel subspace method in [17], which also use the idea of direct channel vector subspace estimation for channel equalization and multiuser detection. As introduced above, our methods first use a linear transformation to process the channel output data, where the desired user’s spreading code information is used, and then, we extract one column vector of the desired user by linear prediction. The methods in [17], however, extract one column vector for all users’ channel matrix by correlation matrix optimization. The extracted mixed signal is then separated by another optimization utilizing the desired user’s code. The zero-forcing criteria is used in both procedures.

Because of the complex operations involved in the correlation matrix optimization procedures in [17], only batch algorithms are available. Although it is not clear whether the second step can be recursively implemented, we find that the first step, i.e., extracting a mixture of one column from all users, is equivalent to multichannel linear prediction. However, two sets of multichannel linear predictions have to be computed in [17] instead of one, as in our method. The procedure of applying desired user’s code first before optimization gives our method some advantages. First, it is computationally much simpler and is easy for adaptive implementation. Second, it may be more robust to optimization errors by using the desired user’s code information first before optimization.

Then, we compare our MMSE algorithms in Section IV-B with the minimum output energy (MOE) or minimum variance CDMA receivers in [8] and [9]. Our MMSE detectors share the same form of minimizing output energy $||\mathbf{m}^H \mathbf{X}||^2$ under a linear constraint. Our approach, however, is different from those of [8] and [9] in the constraint. In our case, $\mathbf{H}_1$ in the constraint $\mathbf{m}^H \mathbf{H}_1 = 1$ is estimated directly by a multichannel linear prediction, whereas the constraints in [8] and [9] turn out to be a channel estimation by the Capon maximization method.

The MOE method requires computing $\mathbf{C}^+$ (see [8, eq. (50) and Step 1, p. 108]), which could be numerically unstable when the channel lengths become large. For example, when the channel length $L > 20$, the $\mathbf{C}^+$ then has magnitudes that are too large for the algorithm to work. Our method, however, is stable, even under the long channel length condition. Besides, when the channel lengths become large, the detector length and matrix size of the MOE of [8] and [9] (and, thus, the computations) grow much faster than our methods. As pointed out in [12], the constrained estimation method of [9] is not robust under randomly selected channels and noisy conditions. This is not the case for our algorithms. See Section VII for simulation results.

**V. PROPERTIES OF THE PROPOSED ALGORITHMS**

**A. SINR Analysis**

In this subsection, we derive analytical expressions to compare the SINR performance of our method with the optimal trained MMSE detector, where SINR denotes the output signal-to-interference and noise ratio. In order to simplify notations, we assume symbol variance $1$, then $\mathbf{R}$ from (3.7), for any linear detector $\mathbf{f}$, the output SINR is given by

$$\text{SINR} = \frac{\mathbf{f}^H \mathbf{H}_1 \mathbf{H}_1^H \mathbf{f}}{\mathbf{f}^H (\mathbf{R}_{11} - \mathbf{H}_1 \mathbf{H}_1^H) \mathbf{f}}. \quad (5.1)$$

The trained optimal MMSE detector is $\mathbf{f}_{\text{mmse}} = \mathbf{R}_{11}^{-1} \mathbf{H}_1$, and thus [9]

$$\text{SINR}_{\text{mmse}} = \frac{1}{\mathbf{H}_1^H \mathbf{R}_{11}^{-1} \mathbf{H}_1 - 1}. \quad (5.2)$$

For the blind MMSE detector introduced in Section IV-B, i.e., (4.18), if we assume that the estimation $\mathbf{u} = \mathbf{H}_1$, then it is easy to show that

$$\text{SINR}_{\text{brmse}} = \text{SINR}_{\text{mmse}}. \quad (5.3)$$

Hence, theoretically, our blind MMSE multiuser detector converges to the optimal trained MMSE detector.
Now, we analyze the SINR performance of the blind zero-forcing detector of Section IV-A, i.e., (4.12). We again assume exact channel vector subspace estimation $\mathbf{u} = \mathbf{H}_1$ and

$$\mathbf{R}_{e_2} = E\{e_2(n)e_2^H(n)\} = \tilde{\mathbf{H}}_1\tilde{\mathbf{H}}_1^H = \mathbf{R}_{11} - \mathbf{H}_1\mathbf{H}_1^H - \sigma_v^2\mathbf{I}.$$  \hfill (5.4)

Then, (4.12) is equivalent to

$$\min_\mathbf{f} J(\mathbf{n}) = \mathbf{f}^H\mathbf{R}_{e_2}\mathbf{f}, \quad \text{subject to } \mathbf{f}^H\mathbf{H}_1 = 1. \hfill (5.5)$$

The zero-forcing detectors were developed under the noiseless case. In that case, $\mathbf{R}_{e_2}$ is not full rank. Therefore, due to the full column rank assumption of the channel matrix, (5.5) has exact nontrivial solutions that make $J(\mathbf{n}) = 0$. Therefore, $\Sigma_{zf} \to \infty$ when SNR $\to \infty$ as seen from (5.1) and (5.4). Hence, under high SNR, the performance of zero-forcing blind detector converges to the trained optimal MMSE detector.

Under the noisy condition, however, $J(\mathbf{n})$ is usually greater than 0. After some manipulations with Lagrange multipliers, we obtain from (5.5)

$$\mathbf{f} = (\mathbf{H}_1^H\mathbf{R}_{e_2}^{-1}\mathbf{H}_1)^{-1}\mathbf{R}_{e_2}^{-1}\mathbf{H}_1. \hfill (5.6)$$

Substituting (5.6) into (5.1), one obtains

$$\text{SINR}_{zf} = \frac{1}{\mathbf{H}_1^H\mathbf{R}_{e_2}^{-1}\mathbf{R}_{11}\mathbf{H}_1 - 1} = \frac{\mathbf{R}_+^H\mathbf{H}_1}{1 - \mathbf{H}_1^H\mathbf{R}_+^H\mathbf{H}_1}. \hfill (5.7)$$

**Proposition 3:** $\text{SINR}_{zf} \leq \text{SINR}_{mmse}$.

**Proof:** Define $\mathbf{R}_{e_2} = \mathbf{R}_{11} - \sigma_v^2\mathbf{I}$ to be the noiseless correlation. Then, using the matrix inversion lemma, we can readily obtain

$$\mathbf{R}_{e_2}^{-1}\mathbf{H}_1 = (\mathbf{R}_{e_2} - \mathbf{H}_1\mathbf{H}_1^H)^{-1}\mathbf{H}_1 = \frac{\mathbf{R}_+^H\mathbf{H}_1}{1 - \mathbf{H}_1^H\mathbf{R}_+^H\mathbf{H}_1}. \hfill (5.8)$$

Hence, the item in the denominator of (5.7) is

$$\frac{\mathbf{H}_1^H\mathbf{R}_{e_2}^{-1}\mathbf{R}_{11}\mathbf{R}_{e_2}^{-1}\mathbf{H}_1}{\|\mathbf{H}_1^H\mathbf{R}_{e_2}^{-1}\mathbf{H}_1\|^2} = \frac{\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{R}_{11}\mathbf{R}_+^H\mathbf{H}_1}{\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{H}_1\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{H}_1}. \hfill (5.9)$$

Note that the corresponding item in $\text{SINR}_{mmse}$ (5.2) is

$$\frac{1}{\mathbf{H}_1^H\mathbf{R}_{11}^{-1}\mathbf{H}_1}. \hfill (5.10)$$

In order to compare their magnitude, we subtract (5.9) from (5.8)

$$\frac{\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{R}_{11}\mathbf{R}_+^H\mathbf{H}_1}{\|\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{H}_1\|^2} - \frac{1}{\mathbf{H}_1^H\mathbf{R}_{11}^{-1}\mathbf{H}_1} = \frac{\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{R}_{11}\mathbf{R}_+^H\mathbf{H}_1 - (\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{H}_1)^2}{(\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{H}_1)^2(\mathbf{H}_1^H\mathbf{R}_+^H\mathbf{H}_1)^2}. \hfill (5.11)$$

Define the singular value decomposition of $\mathbf{R}_{11}$ as $\mathbf{R}_{11} = \mathbf{USU}^H$, where $\mathbf{S}$ is diagonal with eigenvalues $s_i$. Then $\mathbf{R}_+^H = \mathbf{U}(\mathbf{S} - \sigma_v^2\mathbf{I})^+\mathbf{U}^H$. Let the vector $\mathbf{H}_1^H\mathbf{U} = \mathbf{a}^H$ with vector elements $a_i$. Then, the numerator of (5.10) equals

$$a_i^H(\mathbf{S} - \sigma_v^2\mathbf{I})^+\mathbf{S}(\mathbf{S} - \sigma_v^2\mathbf{I})^+\mathbf{a}^H - \mathbf{a}^H(\mathbf{S} - \sigma_v^2\mathbf{I})^+\mathbf{a}(\mathbf{S} - \sigma_v^2\mathbf{I})^+a_i$$

$$= \sum_i |a_i|^2 \frac{s_i}{s_i - \sigma_v^2} \sum_j |k_j|^2 \frac{1}{s_j} - \sum_i |a_i|^2 \frac{1}{s_i - \sigma_v^2} \sum_j |a_j|^2 \frac{1}{s_j - \sigma_v^2}$$

$$= \sum_i \sum_j |a_i|^2 |a_j|^2 \frac{1}{s_i - \sigma_v^2} \frac{1}{s_j - \sigma_v^2}$$

$$\cdot \left( \frac{s_i}{s_i} - 1 \frac{1}{s_i - \sigma_v^2} \right)$$

$$= \sum_i \sum_{j < i} |a_i|^2 |a_j|^2 \frac{1}{(s_i - \sigma_v^2)(s_j - \sigma_v^2)} \frac{1}{s_i} - \frac{1}{s_j} + \frac{1}{s_i - \sigma_v^2} \frac{1}{s_j - \sigma_v^2}.$$  \hfill (5.12)

Hence, the proposition is true.

**B. Propagation Delay**

Thus far, we have assumed correct initial timing recovery for the desired user. Next, we will consider how to estimate this timing information and will analyze the performance under timing mismatch. We have assumed without loss of generality that the delay $d_i$ takes on values from the set $[0, 1, \ldots, L_e - 1]$. With the desired delay estimation $d_i = 0$, we applied the transformation matrix $\mathbf{T}$ to the data vector $\mathbf{y}(n)$ to get rid of the symbol $b(n - d_f)$. Here, $\mathbf{T}$ is chosen according to the delay $d_i = 0$, i.e., $\mathbf{T} = \mathbf{0}$ (see Proposition 1). If the timing estimation is incorrect, then $\mathbf{T}\mathbf{y}(n)$ still contains $b(n - d_f)$. Therefore, $\mathbf{T}$ does not get rid of any symbol contents. Then, the prediction problem (3.6) will converge to $\mathbf{e}_i(n) = 0$. Therefore, the initial timing estimation can be performed by the following maximization procedure:

$$\hat{d}_i = \max_{0 \leq d_i \leq L_e - 1} \|\mathbf{e}_i(n)\|.$$  \hfill (5.13)

This procedure can be incorporated into either the adaptive algorithms or the batch algorithms.
On the other hand, if the channel is overestimated, or there are some zero coefficients in the head or the tail of the channel response, then our methods are robust to timing estimation errors, which is similar to [12]. For example, if there are \( K \) zeros appended to the tail of the channel response, i.e., channel length is overestimated by \( K \) chips, then our methods are robust to backward timing estimation error up to \( K \) chips. On the other hand, we can also purposefully shift \( d_i \) by, e.g., \( K/2 \) chips. Then, our methods will be robust to timing estimation errors in both directions by \( K/2 \) chips. However, since such a treatment will be similar to [12], it is omitted here.

VI. SIMULATIONS

Simulation examples are presented in this section to illustrate the effectiveness of the proposed algorithms in comparison with other existing algorithms. In all of the simulations, the channel response of each user is randomly generated [6], [12] by

\[
g(t) = \sum_{q=1}^{L_d} \alpha_q p(t - \tau_q)
\]

where

- \( L_d \): total number of multipaths;
- \( \tau_q \): associated delay of the \( q \)-th path;
- \( \alpha_q \): attenuation of the \( q \)-th path;
- \( p(t) \): raised-cosine pulse function.

\( g(t) \) is then sampled and truncated to the length \( L_g \). The user delay \( d_i \), the multipath delay \( \tau_q \), and the number of multipath components \( L_d \) are uniformly distributed within [1 \( L_c \)], \([0, (L_h - 1)T_s]\), and [1 \( 30\)], respectively. We use Gold sequence of length \( L_c = 31 \). There are altogether \( J = 10 \) users, unless otherwise stated. The first user is the desired user. All other users have the same signal power, which is different from that of the desired user. The near–far ratio is defined as \( 20 \log_{10} A_i/A_1 \) for \( i \neq 1 \). All input symbols are drawn from a BPSK constellation and then multiplied by various magnitude factors to generate the near–far situations. The signal-to-noise ratio (SNR) is defined as

\[
\text{SNR} = \frac{E[|x(n) - v(n)|^2]}{E[|v(n)|^2]}
\]

A. Long Channels

In this example, we test the performance of our algorithms in long channels. Channel length \( L_g = 30 \) chips. \( N = 2 \), and \( L_h = 3 \). The number of antennas is \( L = 1 \). Therefore, the required total detector length is \( NLL_c = 62 \), which is used for our proposed algorithms. (Note that when varying user number \( J \), \( N \) is also varied.) The detection delay of the proposed algorithms is \( d_f = 0 \).

We first compare the performance of our batch algorithms, i.e., the zero-forcing batch algorithm (ZF) (Algorithm 1, Section IV-A) and the MMSE batch algorithm (MMSE) (Algorithm 3, Section IV-B). We use the subspace algorithm of [6] (SS) and linear prediction-based algorithm of [12] (LP) for comparison. The detector length of 93, which is required by the LP algorithm, was used for all simulations with the above two algorithms. This length surpasses the length 62 of our proposed algorithms and thus gives better performance to these two algorithms than the length 62 case. Note that the MOE algorithms of [8] and [9] will not work in this simulation due to the long channel length. The performance comparisons under various SNR, various data points, various user numbers, and various near–far ratios are shown in Figs. 2–5, respectively. All signal-to-interference-and-noise-ratios (SINRs) [8] are calculated by averaging 100 runs.

The ZF algorithm (Algorithm 2, Section IV-A) and MMSE adaptive algorithm (MMSE) (Algorithm 4, Section IV-B) are then compared with the LMS-type LP adaptive algorithm in [12]. Learning step \( \mu \) for the multichannel linear prediction is 0.0001. The results are presented in Fig. 6.

From the simulations, we see that our algorithms outperform the subspace algorithm and the linear prediction algorithm even when the detector length is shorter for our new algorithms. The MMSE algorithms are slightly better than the zero-forcing algorithms. Note that for simplicity, in rank estimation of the batch algorithm, we used empirically selected thresholds instead of using the AIC or MDL. The results would have been similar or better if the AIC or MDL were used.
B. Short Channels

In this example, we compare the performance of the batch algorithms under short channels. The channel length $L_g = 6$. There are altogether 15 users. All other parameters are identical to the previous example. We compare our algorithms with the MOE algorithm in [8] and [9] as well as the SS algorithm of [6] and the LP-based algorithm of [12]. The detector lengths are 93 for all five algorithms. We use 500 symbols. The results are plotted in Fig. 7. It shows that our MMSE and ZF algorithms have better performance.

VII. Conclusion

In this paper, we consider blind joint multiuser detection and channel equalization for CDMA system with multipath channels. We use linear prediction to estimate channel matrix vector spaces. After that, both the zero-forcing detector and the MMSE detector can be obtained without explicit channel estimation. Both batch algorithms and adaptive algorithms are obtained. These new methods have better performance compared with some other typical methods.

REFERENCES


Xiaohua Li received both the B.S. and M.S. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1992 and 1995, respectively. In June 2000, he received the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH.

Since July 2000, he has been an Associate Professor with the Department of Electrical Engineering, State University of New York, Binghamton. From 1995 to 1996, he worked as an Electrical Engineer with the Shanghai Jiao Tong University and Shanghai Medical Instruments Corporation. His work focused on digital signal processor-based digital hearing aids and multichannel cochlear implants. His current research includes adaptive and array signal processing, blind channel identification and equalizations for wireless communications, and single and multiuser detection for CDMA systems.

Dr. Li was awarded the Excellent Graduates of Shanghai and Chinese Instrument Society Scholarship.

H. Howard Fan (SM’90) received the B.S. degree from Guizhou University, Guiyang, China, in 1976 and the M.S. and Ph.D. degrees from the University of Illinois, Urbana, in 1982 and 1985, respectively. In June 2000, he received the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH.

From 1977 to 1978, he worked as a Research Engineer with the Provincial Standard Laboratory and Bureau of Guizhou Province, Guiyang. In 1978, he joined the Graduate School of the University of Science and Technology of China and then transferred to the University of Illinois, where he was a Teaching and Research Assistant from 1982 to 1985. He joined the Department of Electrical and Computer Engineering, University of Cincinnati, Cincinnati, OH, in 1985 as an Assistant Professor and is now a Professor. His research interests are in the general fields of systems and signal representation and reconstruction, adaptive signal processing, signal processing for communications, and system identification. He spent six months in 1994 with the Systems and Control Group, Uppsala University, Uppsala, Sweden, as a Visiting Researcher.

Dr. Fan received the first ECE Departmental Distinguished Progress in Teaching Excellence Award from the University of Cincinnati in 1987. He served as an Associate Editor of the *IEEE TRANSACTIONS ON SIGNAL PROCESSING* from 1991 to 1994. He is a member of Tau Beta Pi and Phi Kappa Phi.