An Adaptive Linear Prediction Algorithm for Joint Blind Equalization and Blind Multiuser Detection in CDMA

H. Howard Fan and Xiaohua Li
Department of ECECS, University of Cincinnati
Cincinnati, OH 45221-0030
h.fan@uc.edu, xiaohua@ececs.uc.edu

Abstract

In this paper an adaptive algorithm based on linear prediction is presented for joint blind equalization and multiuser detection for asynchronous CDMA systems in multipath propagation. The linear prediction is computationally efficient and numerically robust. It is shown that the new method is insensitive to estimation error of propagation delay, or chip timing. Simulation results show that the proposed method is also near-far resistant, and compare favorably to many existing methods.

1. Introduction

High speed code division multiple access (CDMA) systems suffer from the problem of interchip interference (ICI), which may arise from the common phenomenon of multipath propagation, for example. To achieve reliable communication in these situations, channel equalization is necessary to eliminate ICI. In addition, multiuser detection and multiple access interference (MAI) rejection are simultaneously needed in a CDMA system. Therefore, joint multiuser detection and channel estimation/equalization are essential to practical high speed CDMA systems. A few kinds of blind methods have been proposed for the joint blind multiuser detection and blind channel estimation/equalization problem in a CDMA system. The first kind is subspace based methods [1], [2]. These algorithms all require singular value decomposition (SVD) or eigenvalue decomposition (EVD) of some form of the data correlation matrix. The computational burdens are prohibitively high. 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 [3], [4] which results in computationally efficiently adaptive algorithms. 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 [2], [3]. Furthermore, the performance of the algorithm in [3] critically depends on the first tap of the channel response.

In the single user blind equalization literature, in addition to the subspace based methods and cost function based methods, there also exist methods based on linear prediction [5]. These methods are computationally simpler than the subspace based methods, and are also easily amenable to adaptive implementation. In addition, these methods are robust to channel order estimation and channel estimation errors. In a multiuser CDMA system, however, these methods would fail since no MAI removal had been taken into consideration in [5].

An LP approach in blind synchronous CDMA detection has recently been proposed in [6]. In this paper, we extend that approach to asynchronous case, and present a computationally more reliable method. We assume that only the desired user's spread code is available. The new method can be both implemented as a batch algorithm and efficiently implemented as an adaptive algorithm. We show that the new method is insensitive to timing estimation error and outperforms many existing algorithms.

2. Problem Formulation

Figure 1 is a block diagram of a discretized CDMA receiver for the jth user. There are altogether J users in the entire system. \( b_j(t) \) is the jth user's symbol sequence. \( c_j(k) \) is the jth user's spreading code. \( g_j(k) \) is the jth (multipath) channel impulse response. And \( n(k) \) is the additive noise. For blind
where $L_h$ is related to the length of $h_j(k)$ and the delay $d_j$. Now furthermore stacking up $x_j(k), x_j(k-1), \cdots, x_j(k-L_w+1)$ and using (8), we have

$$
\begin{bmatrix}
    x_j(k) \\
    \vdots \\
    x_j(k-L_w+1)
\end{bmatrix}
= 

\begin{bmatrix}
    h_j^{(d_j)}(0) & \cdots & h_j^{(d_j)}(L_h-1) \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & h_j^{(d_j)}(L_h-1)
\end{bmatrix}
\begin{bmatrix}
    b_j(k) \\
    \vdots \\
    b_j(k-L_h-L_w+2)
\end{bmatrix}
\tag{9}
$$

Or, again with obvious vector and matrix notations, (9) can be written as

$$X_j(k) = \mathcal{H}_j^{(d_j)} b_j(k) \tag{10}$$

Ignoring the noise $n(k)$ for the time being, the received signal vector $X(k)$ can then be expressed as,

$$X(k) = \sum_{j=1}^{J} X_j(k) = \left[ \mathcal{H}_1^{(d_1)} \cdots \mathcal{H}_J^{(d_J)} \right] \begin{bmatrix} b_1(k) \\ \vdots \\ b_J(k) \end{bmatrix} \tag{11}$$

Define

$$\mathcal{H}^{(d_j)} = [\mathcal{H}_1^{(d_1)} \cdots \mathcal{H}_J^{(d_J)}] \tag{12}$$

$\mathcal{H}^{(d_j)}$ is of dimension $L_wL_c \times J(L_h + L_w - 1)$. For $L_c > J$ (no overloading), $L_w$ can be chosen such that $L_wL_c \geq J(L_h + L_w - 1)$.

### 3. Joint Blind Equalization And Multiuser Detection

Without loss of generality, we assume that the first user’s code is known ($j = 1$) and we will estimate the first user’s symbols. We assume that the symbols $b_j(k)$ are uncorrelated in time, with variance $A_j$, for all $j$. $b_i(k)$ and $b_j(k)$, $i \neq j$, are also uncorrelated. We assume that the timing of the desired user $d_1$ is known through timing recovery. Blind timing estimation is discussed in [8]. But other $d_j, j \neq 1$, are unknown and are randomly distributed between 0 and $L_c - 1$. Without loss of generality, we can set $d_1 = 0$. Now we need to express matrix $\mathcal{H}_1$ in terms of the spreading code $c_1(k)$ and the channel impulse response $g_1(k)$. Suppose the actual channel coefficients are from $g_j(0)$ to
\(\text{But for } j \neq 1, d_j \neq 0, \text{ we have (without loss of generality) for } d_j > 0,\)

\[
h_j^{(d_j)}(i) \triangleq C_{ji}^{(d_j)} g_{j}, \ i = 0, 1, 2 \quad (16)
\]

where \(C_{ji}^{(d_j)}\) are upward shifted versions of \(C_{ji}^{(0)}\).

Obviously, \(L_h = 3\) if \(d_j > 1\). Therefore, for a random delay \(d_j\), we may have \(L_h = 3\). Thus in contrast to (15), we may have for \(j \neq 1\), three non-zero columns \(h_j^{(d_j)}(0), h_j^{(d_j)}(1), \text{ and } h_j^{(d_j)}(2)\) in \(H_j^{(d_j)}\).

**A. Linear Transformation**

In [6] we left multiplied \(C_{11}^{-1}\) to the data vector \(X(k)\).

Due to (15) we made the second column of \(C_{11}^{-1} H_1^{(0)}\) having only one block of \(L_c\) non-zero entries but not in the corresponding columns of \(C_{11}^{-1} H_1^{(0)}, i \neq 1\). This difference enabled us to eliminate MAI in the subsequent LP step. In this paper we choose \(T\), instead of \(C_{11}^{-1}\), to achieve the same objective but with much reduced computation. Suppose we would like to create a column which contains only \(L_c\) non-zero entries in the \((L_1 + 1)\)th column of \(TH_1\) where \(L_1 \geq 1\) is a designer chosen integer. Due to the structure of \(H_1\) as given in (15), we only need to annihilate one block of \(L_c\) non-zero entries, say \(h_1^{(0)}(1)\), in the \((L_1 + 1)\)th column of \(H_1\). The following proposition achieves this objective.

**Proposition 1:** The following matrix \(T\) annihilates the \((L_1)\)th block row \(h_1^{(0)}(1)\) in the \((L_1 + 1)\)th column of \(H_1^{(0)}\) and leaves other rows unchanged.

\[
T = \begin{bmatrix}
I_{L_1-1} & T_1 & T_2 \\
T_1 & I_1 & 0 \\
0 & I_{L_2} & 0
\end{bmatrix}, \quad (17)
\]

where \(L_2 = L_w - L_1 - 1\), \(I_k\) is \(k \times k\) identity matrix, and the \(L_c \times 2L_c\) full row-rank matrix \([T_1 \ T_2]\) consists of the left singular vectors corresponding to the zero singular values of the matrix \([C_{11}^{(0)} \ C_{10}^{(0)}]^T\).

**Proof:** See [8].

Applying \(T\) to other channel matrices \(H_j^{(d_j)}, j \neq 1\), we have the similar results except the corresponding entries of the \((L_1 + 1)\)th column \(T_1 h_j^{(d_j)}(1) + T_2 h_j^{(d_j)}(0), j \neq 1\), will be non-zero if we assume that \(C_{11}(z)\) and \(C_{j1}(z)\) have no common zeros where \(j = 2, \ldots, J\) and \(C_{11}(z) = c_1(L_c - 1)z^{L_c-1} + \cdots + c_1(1)z + c_1(0)\). See [8] for details.

**B. Linear Prediction**

Since \(c_1(k)\) is known to the first user, we can calculate \(T\) beforehand. Left multiply \(T\) to the received signal vector \(X(k)\). Using (12), (15), and Proposition
1, we have

\[ T \mathcal{X}(k) = T[H^{(d_1)}_1, H^{(d_2)}_2, \ldots, H^{(d_J)}_J] \begin{bmatrix} b_1(k) \\ \vdots \\ b_J(k) \end{bmatrix} \]

where the submatrix \( H^{(d_1)}_1 \) has only one non-zero block of rows in the \((L_1 + 1)\)th column, while all other submatrices have two non-zero blocks of rows in the corresponding \((L_1 + 1)\)th columns due to Proposition 1. Decompose \( T \mathcal{H}^{(d_j)} \) as follows

\[ T \mathcal{H}^{(d_j)} = \begin{bmatrix} Q_1 \\ Q_0 \\ Q_2 \end{bmatrix} \]

(18)

where \( Q_0 \) has \( L_c \) rows, \( Q_1 \) has \( L_1 L_c \) rows and \( Q_2 \) has \( L_2 L_c \) rows. Since \( Q_1 \) has \( L_1 \) blocks of \( L_c \) rows, \( Q_0 \) consists of the \((L_1 + 1)\)th block of \( L_c \) rows of \( T \mathcal{H}^{(d_j)} \).

Due to the structure of (18), the matrix \( \begin{bmatrix} Q_1 \\ Q_0 \\ Q_2 \end{bmatrix} \) has one and only one zero column at the \((L_1 + 1)\)th column of the first user’s block. Striking this zero column out, the full column rank of the matrix \( \begin{bmatrix} Q_1 \\ Q_0 \\ Q_2 \end{bmatrix} \) can be guaranteed under certain conditions [8] which include \( L_1 \geq \frac{L_c + 1}{L_c - 1} \) and \( L_2 \geq \frac{2L_c}{L_c - 1} \). Note that in the asynchronous case, this choice of \( L_1 \) is sufficient, but not necessary. A necessary condition is

\[ L_1 \geq \frac{J - 1}{L_c - 1}. \]

(19)

Now define an \( L_c \) dimensional two-sided prediction error vector

\[ \varepsilon(k) = \begin{bmatrix} -P_1 & I & -P_2 \end{bmatrix} T \mathcal{X}(k) \]

(20)

where \( P_1, I \) and \( P_2 \) are \( L_c \times L_1 L_c, L_c \times L_c, L_c \times L_2 L_c \) respectively. The derivation now is the same as that of [6]. The above full column rank result ensures that the matrix \( Q_0 - \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \) has only one non-zero column at the \((L_1 + 1)\)th column of the first user’s block, which yields

\[ \begin{bmatrix} -P_1 & I & -P_2 \end{bmatrix} T \begin{bmatrix} H^{(d_1)}_1 \\ H^{(d_2)}_2 \\ \vdots \\ H^{(d_J)}_J \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \\ H^{(0)}_1 \\ 0 & \cdots & 0 \end{bmatrix} \]

\[ \text{first block} \]

(21)

with \( d_1 = 0 \). In other words,

\[ \begin{bmatrix} -P_1 & I & -P_2 \end{bmatrix} T \mathcal{X}(k) = b_1(k - L_1) h^{(0)}_1(0). \]

(22)

It can also be shown that

\[ E \{ \varepsilon(k) \varepsilon^H(k) \} = A_1 b_1^{(0)}(0) b_1^{(0)H}(0) \]

\[ = A_1 C^{(0)}_{10} g_1 g_1^H C^{(0)H}_{10}. \]

(23)

Therefore \( h^{(0)}_1(0) \) can be easily estimated. After the estimation of \( h^{(0)}_1(0) \), channel identification is obtained by inverting \( C^{(0)}_{10} \) as indicated by (13). The zero-forcing equalizer and decorrelating detector vector is then

\[ w^H(k) = h^{(0)H}_1(0) \begin{bmatrix} -P_1 & I & -P_2 \end{bmatrix} T \]

(24)

It can be shown [8] that this LP algorithm is robust against positive timing estimation error. To cope with negative error, we define

\[ L_0 = \frac{L_c - L_2}{2}. \]

(25)

Then further shift the data by \( L_0 \) chips in implementing the LP algorithm renders it to be robust against both positive and negative timing estimation errors. See [8] for details.

4. Simulations

Results of simulations of both of the adaptive algorithm and a batch algorithm [8] are presented here. In all of the simulations, the channel response of each user is randomly generated based on [1]

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

where \( L_d \) is the total number of multipaths, \( \tau_q \) is the associated delay of the \( q \)th path, \( \alpha_q \) is the attenuation of the \( q \)th path, and \( p(t) \) is the 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 \([1, L_c], [0, 3T_d]\) and \([1, 10]\) respectively. We use Gold sequence of length \( L_c = 31 \). \( L_0 \) is chosen to be \( 10 \). \( L_1 \) is set to be \( 1 \) and \( L_2 = 1 \). Therefore the total detector length is \( L_w L_c = 93 \).

In all simulations the further shifted data (by \( L_0 \)) are used. We use the algorithm of [1] (SS Alg), [3] (MV Alg) and [7] (LSS Alg). The performance comparisons under various data points and various timing mismatch are shown in Figures 2-4.

From the simulations, we see that for relatively long channels (7-10 chips) our algorithms outperform the three methods under comparison.
Figure 2. Performance of the batch algorithms versus data length. $L_g = 10$, $J = 5$, Near-far ratio 0dB, SNR 10dB.

Figure 3. Performance of the batch algorithms under timing estimation error. 1000 symbols, $L_g = 7$, $J = 10$, SNR=5dB, near-far ratio 0dB. For LP Alg., $L_0 = 10$.

Figure 4. Performance of the adaptive algorithms versus SNR. 3000 symbols, $L_g = 5$, $J = 10$, Near-far ratio 5dB.

References


