Blind Channel Estimation for Long Code Multiuser CDMA Systems

Zhengyuan (Daniel) Xu and Michail K. Tsatsanis, *Member, IEEE*

***Abstract*—In CDMA systems with long codes, the users' signatures change in each bit period, impeding the estimation of the time-invariant multipath parameters. In this paper, correlation-matching methods are employed to blindly estimate those multipath parameters. For given code sequences, the output correlation matrix (parameterized by the unknown channel coefficients) is compared with its instantaneous approximation. By minimizing the Frobenius norm of the resulting error matrix, the channel parameters can be estimated up to a complex scalar ambiguity. Both batch and adaptive algorithms are derived. Under the assumption of i.i.d. code sequences, identifiability up to a complex scalar ambiguity for each channel is guaranteed, and the asymptotic convergence of the proposed algorithm is established. Furthermore, step-size selection for the adaptive version is investigated. When only the code sequence of the user of interest is available, a single user receiver is also derived. Simulation results verify those claims and provide comparisons with other methods.**

***Index Terms*—CDMA, channel estimation, correlation, long codes.**

I. INTRODUCTION

DIRECT sequence code division multiple access (CDMA) systems have received considerable attention recently in wireless communications due to their improved capacity [9]. Their importance is expected to increase in future wireless systems, given the increasing demand for wireless services.

Many existing CDMA standards (like IS-95 [9]) utilize a spreading waveform that is generated using a PN sequence with a long period. Hence, only a portion of the waveform is used to spread each bit, resulting in a randomization of each bit's signature and rendering standard adaptive interference cancellation techniques nonapplicable, e.g., [11].

It is no surprise that most signal processing research in related channel estimation problems has focused on the more tractable short code case, where the code's length equals the bit duration [1], [14], [22], [23]. In this case, the code repeats at every bit and the interference structure is time invariant. Direct design methods for multiuser detectors have also been pursued in that case [6], [13], [15], [18], [25], and their performance has been analyzed in [16] and [30]. Recent research efforts have addressed the long code case and contributed to the design of new blind receivers suitable for such systems [24], [31]. In these approaches, channel parameters are first estimated, and receivers can then be constructed.

In this paper, we focus on channel estimation problems from a different point of view by employing correlation matching techniques. Moment matching approaches have been extensively analyzed in the past [17] and have been applied to a multitude of problems in blind identification [5], [28], detection and estimation [8], and fractionally spaced equalization [7]. Extensions to the time-varying systems have been reported in [27].

In the current setup, the channel parameters are assumed time invariant, but due to the changing codes, the users' signatures are time varying. Hence, given the long codes for different users, we match the output covariance matrix (parameterized by these unknown channel vectors) with instantaneous approximations based on the received data. Through this minimization procedure, closed-form solutions of the channel vectors within a complex scalar ambiguity are obtained.

The contribution of this work is to extend correlation matching techniques to the current CDMA setup, where the bit signatures vary rapidly from bit to bit. Compared with previous subspace-based approaches, the proposed method requires only mild identifiability assumptions and offers better performance for loaded systems.

Both batch and adaptive algorithms are investigated, and their asymptotic performance is studied. It is shown that our estimates for all channels correspondingly converge to their true parameters, even for a heavily loaded system. If only a particular user is of interest, a simplified single user channel estimation method can be employed, where the computations are greatly reduced, and only the long code corresponding to that user is required.

The performance of our different methods is compared with the subspace-based approach [31] based on simulation results. It turns out that the proposed multiuser channel estimation methods (batch and adaptive versions) exhibit superior performance to both the proposed single-user channel estimation algorithms and the subspace method [31], especially in a heavily loaded system. The proposed adaptive algorithm is also tested in a Rayleigh fading communication environment, and channel tracking is performed for such a scenario.

The rest of the current paper is organized as follows. In Section II , a discrete-time model for a long code multiuser CDMA system is provided. Section III describes in detail our blind batch and adaptive multiuser channel estimation algorithms, whereas their performance is analyzed in Section IV. When only a single user is of interest, simplified channel estimation methods are

Manuscript received September 25, 1998; revised October 30, 1999. This work was supported in part by the National Science Foundation under Grants NSF-NCR 9706658, NSF-CAREER 9733048, and NSF WIRELESS 9979288 and the Army Research Office under Grant DAAG55-98-1-0224. The associate editor coordinating the review of this paper and approving it for publication was Prof. Chi Chung Ko.

Z. D. Xu is with the Department of Electrical Engineering, University of California, Riverside, CA 92521 USA (e-mail: dxu@ee.ucr.edu).

M. K. Tsatsanis is with the Electrical and Computer Engineering Department, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: mtsatsan@stevens-tech.edu).

Publisher Item Identifier S 1053-587X(00)02360-6.

derived in Section V. Simulation results and comparisons with other method are shown in Section VI, and some conclusions are drawn in Section VII.

II. System Model

In multiuser CDMA systems, different users are assigned distinct code sequences to spread the signal spectrum over the available bandwidth. In IS-95, the code sequence for each user is a mod-2 addition of a periodic code and a long pseudo-random sequence [9]. Therefore, each bit to be transmitted is modulated by a time-varying long code. Due to the excessive length of the code (compared with the spreading factor), it is reasonable to model it as an i.i.d. chip sequence. We will focus on the channel model under such a modulation scheme.

Let us assume that there are M users in a DS-CDMA communication system, and let user j ($j = 1, \cdots, M$) transmit a zero-mean, i.i.d. bit sequence $w_j(n)$ with variance $\sigma^2_{w_j} = E\{\|w_j(n)\|^2\}$. At time n, each bit is spread by an uncorrelated i.i.d. code sequence. Let us define $c_{j,k}(n), n = 0, \cdots, P-1$ to be the spreading code of user j, bit k, with P chips (n is the chip index). Then, the jth user's chip rate, discrete-time transmitted signal can be expressed by the multirate convolution (see [26] for details; cf. [29])

$$s_j(n) = \sum_{k=-\infty}^{\infty} w_j(k) c_{j,k}(n - kP). \tag{1}$$

After passing through a multipath channel, at the chip-rate receiver, the output signal contributed by user j is (see Fig. 1)

$$y_j(n) = \sum_{l=-\infty}^{\infty} s_j(l) g_j(n - \delta_j - l) \tag{2}$$

where $0 \le \delta_j < P$ is the delay in chip periods, and $g_j(n)$ is the discrete-time equivalent channel impulse response that includes the transmitter and receiver filters. By considering (1) and (2) together, we can relate the output $y_j(n)$ to its input $w_j(n)$ as

$$y_j(n) = \sum_{k=-\infty}^{\infty} w_j(k) h_{j,k}(n - kP) \tag{3}$$

$$h_{j,k}(n) = \sum_{m=-\infty}^{\infty} g_j(m) c_{j,k}(n - m - \delta_j) \tag{4}$$

where $h_{j,k}(n)$ is the signature of user j for bit k with chip index n. The overall received signal $y(n)$ is then a superposition of signals from all M users corrupted by AWGN $v(n)$

$$y(n) = \sum_{j=1}^{M} y_j(n) + v(n) \tag{5}$$

where $v(n)$ has zero mean and variance $\sigma_v^2 = E\{\|v(n)\|^2\}$. In practice, communication channels are usually modeled as having finite impulse response. Thus, let the maximum order for all multipath channels $g_j(n)$ be q and known in the sequel (zero elements can be appended if its order is less than q).

To obtain a compact form of our model in an observation interval, let us collect $K = P + q$ samples of $y(n)$ in a $K \times 1$ vector $\boldsymbol{y}(n) = [y(nP), \cdots, y(nP + K - 1)]^T$. Then, from (3)–(5), the received signal $\boldsymbol{y}(n)$ becomes[1]

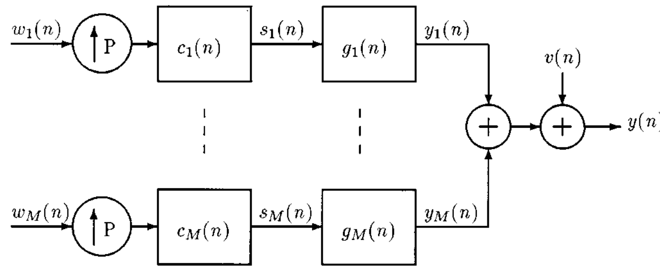


Fig. 1. CDMA system with long spreading codes.

$$\boldsymbol{y}(n) = \sum_{j=1}^{M} \boldsymbol{h}_{j,1}(n) w_j(n) + \sum_{j=1}^{M} \boldsymbol{h}_{j,2}(n-1) w_j(n-1) + \boldsymbol{v}(n) \tag{6}$$

where $\boldsymbol{h}_{j,1}(n) = [0, \cdots, 0, h_{j,n}(1), \cdots, h_{j,n}(K - \delta_j)]^T$ (with δ_j zeros), $\boldsymbol{h}_{j,2}(n-1) = [h_{j,n-1}(P + 1 - \delta_j), \cdots, h_{j,n-1}(K), 0, \cdots, 0]^T$ [with $K - (q + \delta_j)$ zeros] are $K \times 1$ vectors corresponding to signatures of $w_j(n)$, $w_j(n-1)$ at time n, respectively. According to (4), these signatures can be expressed as

$$\boldsymbol{h}_{j,1}(n) = \boldsymbol{C}_{j,1}(n)\boldsymbol{g}_j, \qquad \boldsymbol{h}_{j,2}(n-1) = \boldsymbol{C}_{j,2}(n-1)\boldsymbol{g}_j \tag{7}$$

where the $(q + 1) \times 1$ channel vector of user j is

$$\boldsymbol{g}_j = [g_j(0), \cdots, g_j(q)]^T$$

and the $K \times (q + 1)$ code filtering matrices are defined by

$$\boldsymbol{C}_{j,1}(n) = \begin{bmatrix} \boldsymbol{0}_{\delta_j \times (q+1)} \\ \boldsymbol{C}_j(n)(1 : K - \delta_j, :) \end{bmatrix}$$
$$\boldsymbol{C}_{j,2}(n-1) = \begin{bmatrix} \boldsymbol{C}_j(n-1)(P + 1 - \delta_j : K, :) \\ \boldsymbol{0}_{(P-\delta_j) \times (q+1)} \end{bmatrix} \tag{8}$$

$$\boldsymbol{C}_j(n) = \begin{bmatrix} c_{j,n}(0) & & O \\ \vdots & \ddots & c_{j,n}(0) \\ c_{j,n}(P-1) & & \vdots \\ O & \ddots & c_{j,n}(P-1) \end{bmatrix}_{K \times (q+1)} \tag{9}$$

and notation $\boldsymbol{X}(l_1 : l_2, :)$ from Matlab is used to take out those rows from l_1 to l_2 of matrix $\boldsymbol{X}$. By substituting (7) in (6), the output becomes[2]

$$\boldsymbol{y}(n) = \sum_{j=1}^{M} [\boldsymbol{C}_{j,1}\boldsymbol{g}_j w_j(n) + \boldsymbol{C}_{j,2}\boldsymbol{g}_j w_j(n-1)] + \boldsymbol{v}(n). \tag{10}$$

If we define the $K \times K$ correlation matrix of this received data at time n as

$$\boldsymbol{R}_y(n) = E\{\boldsymbol{y}(n)\boldsymbol{y}^H(n)|_{\boldsymbol{C}_{j,\, j=1,\cdots,M}}\} \tag{11}$$

[1]If $\delta_j < q$, then the third term corresponding to the symbol $w_j(n+1)$ will appear. The proposed method can be easily extended. For clarity of presentation, we will ignore this term later following common practice.

[2]We will drop the time indices "n" and "$n-1$" for code matrices $\boldsymbol{C}_{j,1}(n)$, $\boldsymbol{C}_{j,2}(n-1)$, respectively, to simplify the notation.

conditioned on the given code sequences for M users, where superscript "H" denotes complex conjugate transpose (Hermitian); then, from (10)

$$\boldsymbol{R}_y(n) = \sum_{j=1}^{M} \sigma_{w_j}^2 [\boldsymbol{C}_{j,1}\boldsymbol{g}_j\boldsymbol{g}_j^H\boldsymbol{C}_{j,1}^H + \boldsymbol{C}_{j,2}\boldsymbol{g}_j\boldsymbol{g}_j^H\boldsymbol{C}_{j,2}^H] + \sigma_v^2\boldsymbol{I}. \tag{12}$$

In this paper, we will assume in the sequel that these spreading codes and delays for all users are known in developing a multiuser channel estimation algorithm in Section III, or only the spreading code and delay for the user of interest is given in deriving our single-user channel estimation method in Section V. The code acquisition problem, e.g., [21], is beyond the scope of this paper. Therefore,,it is clear that $\boldsymbol{R}_y(n)$ is only parameterized by σ_v^2, $\sigma_{w_j}\boldsymbol{g}_j$. Notice that due to the time-varying feature of code sequences, $\boldsymbol{R}_y(n)$ is also time varying. However, we assume all unknowns $\sigma_{w_j}\boldsymbol{g}_j$ and σ_v^2 are kept constant. Hence, the estimation problem is focused on the retrieval of these unchanged parameters based on the knowledge of $\boldsymbol{R}_y(n)$. We will focus on estimating all $\boldsymbol{g}_j$ next, based on correlation matching techniques [17].

III. Blind Multiuser Channel Estimation

Let $\hat{\boldsymbol{R}}_y(n)$ denote some estimator of $\boldsymbol{R}_y(n)$, and let

$$\boldsymbol{E}(n) = \boldsymbol{R}_y(n) - \hat{\boldsymbol{R}}_y(n) \tag{13}$$

be the estimation error matrix at time n. To evaluate this estimation error, let us define an index as the square of the Frobenius norm of $\boldsymbol{E}(n)$

$$J(n) = \|\boldsymbol{E}(n)\|_F^2 = \mathrm{tr}[\boldsymbol{E}(n)\boldsymbol{E}^H(n)]. \tag{14}$$

By using the following property of "vec" operation and the trace (see [10, ch. 12])

$$\mathrm{tr}(\boldsymbol{X}_1\boldsymbol{X}_2^T) = \mathrm{vec}^T(\boldsymbol{X}_2)\mathrm{vec}(\boldsymbol{X}_1) \tag{15}$$

(14) becomes

$$J(n) = \mathrm{vec}^H[\boldsymbol{E}(n)]\mathrm{vec}[\boldsymbol{E}(n)]. \tag{16}$$

Then, we can build our cost function as a cumulative error

$$J = \frac{1}{N}\sum_{n=1}^{N} J(n) = \frac{1}{N}\sum_{n=1}^{N} \mathrm{vec}^H[\boldsymbol{E}(n)]\mathrm{vec}[\boldsymbol{E}(n)] \tag{17}$$

where N is the number of bit periods. By minimizing this cost function, all channel parameters could be obtained.

However, two problems will impede our estimation. First, $\hat{\boldsymbol{R}}_y(n)$ needs to be known in (17). Due to the time-varying property of $\boldsymbol{R}_y(n)$, the sample average over the data record (which is usually used for a time invariant system) is not applicable here. Instead, we will use its instantaneous approximation $\boldsymbol{y}(n)\boldsymbol{y}^H(n)$

$$\hat{\boldsymbol{R}}_y(n) = \boldsymbol{y}(n)\boldsymbol{y}^H(n). \tag{18}$$

It may seem that $\hat{\boldsymbol{R}}_y(n)$ is very inaccurate, and hence, $\hat{\boldsymbol{g}}_j$ (the estimate of $\boldsymbol{g}_j$) will not be reliable. However, our cost function J employs all data points and results in surprisingly reliable estimates. This is further supported by our consistency results.

Second, the cost function J in (17) is a fourth-order function of the unknowns $\boldsymbol{g}_j$ [see also (12) and (13)]; thus, its high nonlinearity may lead to difficulties in estimating the channel vector $\boldsymbol{g}_j$. With this in mind, if we define our new unknowns

$$\boldsymbol{D}_j = \sigma_{w_j}^2\boldsymbol{g}_j\boldsymbol{g}_j^H, \qquad x = \sigma_v^2 \tag{19}$$

then (12) is linearly parameterized by $\boldsymbol{D}_j$ and x

$$\boldsymbol{R}_y(n) = \sum_{j=1}^{M}(\boldsymbol{C}_{j,1}\boldsymbol{D}_j\boldsymbol{C}_{j,1}^H + \boldsymbol{C}_{j,2}\boldsymbol{D}_j\boldsymbol{C}_{j,2}^H) + x\boldsymbol{I}. \tag{20}$$

Substituting (20) and (18) in (13), the error matrix becomes

$$\boldsymbol{E}(n) = \sum_{j=1}^{M}(\boldsymbol{C}_{j,1}\boldsymbol{D}_j\boldsymbol{C}_{j,1}^H + \boldsymbol{C}_{j,2}\boldsymbol{D}_j\boldsymbol{C}_{j,2}^H) + x\boldsymbol{I} - \boldsymbol{y}(n)\boldsymbol{y}^H(n). \tag{21}$$

To evaluate vec$[\boldsymbol{E}(n)]$ required in our cost function (17), another property of "vec" can be employed (see [10, ch. 12])

$$\mathrm{vec}(\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3) = (\boldsymbol{X}_3^T \otimes \boldsymbol{X}_1)\mathrm{vec}(\boldsymbol{X}_2) \tag{22}$$

where "$\otimes$" represents the Kronecker product. After defining new vectors $\boldsymbol{d}_j$, $\boldsymbol{d}$ (of length $(q+1)^2$ and $M(q+1)^2$, respectively)

$$\boldsymbol{d}_j = \mathrm{vec}(\boldsymbol{D}_j), \qquad \boldsymbol{d} = [\boldsymbol{d}_1^T, \cdots, \boldsymbol{d}_M^T]^T \tag{23}$$

a compact form for vec$[\boldsymbol{E}(n)]$ can be obtained from (21)

$$\mathrm{vec}[\boldsymbol{E}(n)] = \boldsymbol{Q}\boldsymbol{d} + \mathrm{vec}(\boldsymbol{I})x - \mathrm{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)] \tag{24}$$

where $\boldsymbol{Q}$ is a code-related matrix of dimension $K^2 \times M(q+1)^2$ with time index dropped for notational convenience

$$\boldsymbol{Q} = [\boldsymbol{Q}_1, \cdots, \boldsymbol{Q}_M], \qquad \boldsymbol{Q}_j = \boldsymbol{C}_{j,1}^* \otimes \boldsymbol{C}_{j,1} + \boldsymbol{C}_{j,2}^* \otimes \boldsymbol{C}_{j,2} \tag{25}$$

and "$*$" denotes complex conjugate. Substituting (24) in (16) first, we have

$$J(n) = \{\boldsymbol{Q}\boldsymbol{d} + \mathrm{vec}(\boldsymbol{I})x - \mathrm{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}^H \cdot \{\boldsymbol{Q}\boldsymbol{d} + \mathrm{vec}(\boldsymbol{I})x - \mathrm{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}. \tag{26}$$

Therefore, our cost function becomes

$$J = \frac{1}{N}\sum_{n=1}^{N}\{\boldsymbol{Q}\boldsymbol{d} + \mathrm{vec}(\boldsymbol{I})x - \mathrm{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}^H \cdot \{\boldsymbol{Q}\boldsymbol{d} + \mathrm{vec}(\boldsymbol{I})x - \mathrm{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}. \tag{27}$$

Thus, we arrive at a quadratic function of new unknowns by overparameterizing the problem by (19). According to our definition in (19), each $\boldsymbol{D}_j$ bears channel information of user j and should be a rank one matrix. It has a unique maximum eigenvalue $\sigma_{w_j}^2\|\boldsymbol{g}_j\|^2$ with corresponding eigenvector $(\boldsymbol{g}_j/\|\boldsymbol{g}_j\|)e^{i\phi_j}$, where "$i$" is the imaginary unit, and ϕ_j is the phase ambiguity. Taking into account both the norm $\|\boldsymbol{g}_j\|$ and the phase ϕ_j, we can then conclude that this eigenvector is our estimated channel vector within a complex scalar ambiguity or, alternatively, is the normalized channel vector within a phase ambiguity. Following this idea, we will focus on estimating $\boldsymbol{D}_j$ for all M users based on our cost function. Two approaches with different demands on computations will be derived in the sequel.

A. Batch Algorithm

Notice that our cost function (27) is a quadratic function of $\boldsymbol{d}$ and x. If we minimize this cost function, a unique closed-form solution can be obtained. To find the minimum solution of (27), it is sufficient to differentiate it with respect to x and $\boldsymbol{d}$, respectively, and set these derivatives equal to zero. Similarly, as (23), we denote our solution as $\hat{x}_N$ and $\hat{\boldsymbol{d}}_N$

$$\hat{\boldsymbol{d}}_N = \left[\hat{\boldsymbol{d}}_1^T, \cdots, \hat{\boldsymbol{d}}_M^T\right]^T \tag{28}$$

which contains estimates for all our unknown parameters. It is shown in Appendix A that the estimate $\hat{\boldsymbol{d}}_N$ satisfies

$$\boldsymbol{T}_N \hat{\boldsymbol{d}}_N = \boldsymbol{t}_N \tag{29}$$

with a $M(q+1)^2 \times M(q+1)^2$ square matrix

$$\boldsymbol{T}_N = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{Q}^H\boldsymbol{Q}) - \frac{1}{KN^2}\sum_{n=1}^{N}\boldsymbol{b}(n)\sum_{n=1}^{N}[\boldsymbol{b}(n)]^H \tag{30}$$

and a $M(q+1)^2 \times 1$ vector

$$\boldsymbol{t}_N = \frac{1}{N}\sum_{n=1}^{N}\{\boldsymbol{Q}^H \text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\} - \frac{\sum_{n=1}^{N}[\boldsymbol{y}^H(n)\boldsymbol{y}(n)]}{KN^2}\sum_{n=1}^{N}\boldsymbol{b}(n) \tag{31}$$

where

$$\boldsymbol{b}(n) = \text{vec}(\boldsymbol{H}), \qquad \boldsymbol{H} = [\boldsymbol{H}_1, \cdots, \boldsymbol{H}_M]$$
$$\boldsymbol{H}_j = \boldsymbol{C}_{j,1}^H\boldsymbol{C}_{j,1} + \boldsymbol{C}_{j,2}^H\boldsymbol{C}_{j,2} \tag{32}$$

and $\boldsymbol{C}_{j,1}$, $\boldsymbol{C}_{j,2}$ are given by (8) and (9) with assumed available delays δ_j for $j = 1, \cdots, M$. Notice that all code matrices depend on time, but the time index is dropped for the sake of notational convenience, and hence, $\boldsymbol{H}$ and $\boldsymbol{Q}$ are also time varying. Therefore, (29) has $M(q+1)^2$ unknown parameters in $\hat{\boldsymbol{d}}_N$ under our assumption and $M(q+1)^2$ equations. $\hat{\boldsymbol{d}}_N$ can be uniquely solved as

$$\hat{\boldsymbol{d}}_N = \boldsymbol{T}_N^{-1}\boldsymbol{t}_N \tag{33}$$

as long as the matrix $\boldsymbol{T}_N$ is nonsingular, as will be discussed in Section IV-A. According to (28), our estimates $\hat{\boldsymbol{d}}_j$ can be obtained by taking out corresponding elements of $\hat{\boldsymbol{d}}_N$, and the reverse operation of the vec function can be performed to obtain $\hat{\boldsymbol{D}}_j$. The major computational complexity of this method lies in the matrix inversion that requires $O(M^3(q+1)^6)$ multiplications. This may be computationally prohibitive when M is large. In Section III-B, we will derive an adaptive version of this method to reduce the computational cost.

B. Adaptive Algorithm

Considering the cost function $J(n)$ at time n in (26), we can formulate the following LMS type recursion for $\boldsymbol{d}$ with step size μ

$$\boldsymbol{d}^{(n+1)} = \boldsymbol{d}^{(n)} - \mu\nabla_{\boldsymbol{d}^H} J(n). \tag{34}$$

In (34), $\nabla_{\boldsymbol{d}^H} J(n)$ is a function of $\boldsymbol{d}^{(n)}$ and $x^{(n)}$ and can be computed from (26)

$$\nabla_{\boldsymbol{d}^H} J(n) = \boldsymbol{Q}^H\boldsymbol{Q}\boldsymbol{d}^{(n)} + \boldsymbol{b}(n)x^{(n)} - \boldsymbol{Q}^H\text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]. \tag{35}$$

In order to eliminate $x^{(n)}$, which is a nuisance parameter in our estimation problem, we substitute it by [see (55) in Appendix A]

$$\hat{x}^{(n)} = \frac{1}{K}[\boldsymbol{y}^H(n)\boldsymbol{y}(n) - \boldsymbol{b}^H(n)\boldsymbol{d}^{(n)}] \tag{36}$$

which minimizes $J(n)$ at time n. Based on (34)–(36), $\boldsymbol{d}$ can be updated by

$$\boldsymbol{d}^{(n+1)} = \boldsymbol{d}^{(n)} - \mu\left[\boldsymbol{Q}^H\boldsymbol{Q} - \frac{1}{K}\boldsymbol{b}(n)\boldsymbol{b}^H(n)\right]\boldsymbol{d}^{(n)} - \mu\left\{\frac{1}{K}\boldsymbol{y}^H(n)\boldsymbol{y}(n)\boldsymbol{b}(n) - \boldsymbol{Q}^H\text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\right\} \tag{37}$$

where $\boldsymbol{Q}$, $\boldsymbol{b}(n)$ are given by (25) and (32). Similarly, for all $j = 1, \cdots, M$, $\boldsymbol{D}_j^{(n+1)}$ can be reconstructed from $\boldsymbol{d}_j^{(n+1)}$, which is a part of $\boldsymbol{d}^{(n+1)}$.

Once $\boldsymbol{D}_j$ is found either via the batch algorithm or by the adaptive implementation, SVD on $\boldsymbol{D}_j$ can be performed to obtain its eigenvector corresponding to the unique maximum eigenvalue, which is our estimated normalized channel vector for user j within a phase ambiguity. This is not computationally demanding because $\boldsymbol{D}_j$ is a $(q+1)\times(q+1)$ small size matrix for moderate channel order q. Alternatively, we may employ other subspace tracking methods, e.g., [3], [4], and [20] to track this eigenvector. This adaptive method requires on the order of $K^2M(q+1)^2$ multiplications for the recursion and $M(q+1)^3$ computations for the SVD operation.

Our solution is derived either in (33) or from recursion (37) for the multiuser CDMA communication system. We may wonder if (33) is indeed useful or, equivalently, if matrix $\boldsymbol{T}_N$ is nonsingular, and moreover, if this solution and the adaptive one in (37) converge to the channel parameters as $N \to \infty$. By the same token, the choice of the step size in (37) is also related to those questions and will be discussed next.

IV. PERFORMANCE ANALYSIS

This section will focus on the investigation of properties enjoyed by the proposed methods as $N \to \infty$. Our solution is obtained either for a particular number (N) of observation vectors from (33) or for a particular number of iterations by (37). As is well-known, the minimization of the quadratic cost function in (27) admits a unique solution, but the question here is whether this solution can guarantee consistency as $N \to \infty$. To answer the identifiability question, we start our asymptotic analysis from (29) and further show that our solution in (33) converges to $\boldsymbol{d}$ as $N \to \infty$.

A. Identifiability

The identifiability of channel parameters depends on the properties of matrix $\boldsymbol{T}_N$. This matrix is determined by the spreading codes. As mentioned before, we assume all code sequences $\boldsymbol{c}_j(n)$ $(j = 1, \cdots, M)$ to be i.i.d. in j and n, taking values from a finite alphabet, with zero mean and variance σ_c^2 and fourth-order moment m_{4c}. They are also independent of both the i.i.d. transmitted bits and the AWGN. Then, from (33), it is shown in Appendix B that as $N \to \infty$, our estimator satisfies

$$\boldsymbol{T}\hat{\boldsymbol{d}}_\infty = \boldsymbol{T}\boldsymbol{d} \tag{38}$$

where $\boldsymbol{T}$ is a constant matrix whose closed-form expression is also derived in Appendix B. Equation (38) tells us that our estimate $\hat{\boldsymbol{d}}_\infty$ will converge to its true value $\boldsymbol{d}$ as long as $\boldsymbol{T}$ is nonsingular. As is seen from Appendix B, this constant matrix $\boldsymbol{T}$ is parameterized by system parameters P, M, q, δ_j and the codes' second order, as well as fourth-order information σ_c^2, m_{4c}. It is not possible to obtain a general proof of nonsingularity of matrix $\boldsymbol{T}$ due to a variety of settings of these parameters. However, these parameters (and, therefore, matrix $\boldsymbol{T}$) are known *a priori*. They do not depend on unknown channel coefficients and the transmitted bits. For different combinations of these parameters, we can check the rank of $\boldsymbol{T}$ by computer beforehand. From our numerical test, we observe that for a large range of possible P, e.g., up to 256, M (with $M \le P$), q, e.g., 1 to 20, and binary random spreading codes, $\boldsymbol{T}$ has full rank for synchronous systems ($\delta_j = 0$). If we take into account all possible delays (both zero and nonzero) for M users, then P^M combinations need to be checked because we allow δ_j to vary between $0 \le \delta_j < P$. This nonsingularity of $\boldsymbol{T}$ will lead to the identifiability of matrices $\boldsymbol{D}_j$ for all users.

B. Asymptotic Convergence

Our previous discussion indicates that $\hat{\boldsymbol{D}}_j$ asymptotically converges to $\boldsymbol{D}_j$ for all possible j $(j = 1, \cdots, M)$. Since the normalized channel $\boldsymbol{g}_j/||\boldsymbol{g}_j||$ is the eigenvector of $\boldsymbol{D}_j$ corresponding to its maximum eigenvalue, all channel parameters can be identified up to a complex scalar ambiguity.

C. Convergence in the Mean

Let us turn our attention to the asymptotic behavior of our adaptive algorithm described by (37).

If we introduce $\triangle\boldsymbol{d}^{(n)} = E[\boldsymbol{d}^{(n)}] - \boldsymbol{d}$ to be the bias at time n and use the independence assumption, e.g., [12], and the i.i.d. assumptions of codes, inputs, and noise, then from (37), the difference process abides by the recursion (see Appendix C)

$$\triangle\boldsymbol{d}^{(n+1)} = (\boldsymbol{I} - \mu\boldsymbol{U}) \triangle \boldsymbol{d}^{(n)} \tag{39}$$

where $\boldsymbol{U}$ is a constant matrix parameterized by given system parameters. Its closed-form expression is also derived in Appendix C. Equation (39) implies that the convergence of the proposed adaptive method depends on the eigenvalues of matrix $\boldsymbol{I} - \mu\boldsymbol{U}$. The necessary condition on the step size is then $|1 - \mu\lambda_i| < 1$ $\forall\, i$, where λ_i's are the eigenvalues of $\boldsymbol{U}$ (see [12]). Equivalently, $0 < \mu < 1/\lambda_{\max}$. This analysis provides a way to select our step size μ for a given set of system parameters.

V. Single User Recievers

It is typical in CDMA communication systems with long code sequences to employ single-user receivers as in IS-95. Signals from all other users in the system are treated as interference, which simplifies the receiver design. It is the only feasible solution when information about other users' codes is not available. In other cases, there may be a large number of active users at the same time so that it is not feasible to estimate the channel parameters of all other interfering users. Therefore, it is worthwhile to modify our method for this typical situation.

If there is a large number of users in the system and we focus on the estimation of the channel experienced by a particular user (without loss of generality, we assume user 1 to be the desired user), then due to the i.i.d property of the code sequences of all users, it is reasonable to treat the contribution from other users as a stationary interference with correlation matrix $\boldsymbol{R}_{int}$. Under this assumption, for given $\boldsymbol{C}_1(n)$, the output correlation matrix is defined as

$$\boldsymbol{R}_y(n) = E\{\boldsymbol{y}(n)\boldsymbol{y}^H(n)|_{\boldsymbol{C}_1(n)}\}$$

and can be expressed as

$$\boldsymbol{R}_y(n) = \boldsymbol{C}_{1,1}\boldsymbol{D}_1\boldsymbol{C}_{1,1}^H + \boldsymbol{C}_{1,2}\boldsymbol{D}_1\boldsymbol{C}_{1,2}^H + \boldsymbol{R}_i \tag{40}$$

where $\boldsymbol{R}_i = \boldsymbol{R}_{int} + \sigma_v^2\boldsymbol{I}$ accounts for interference and AWGN, and generally, it is not diagonal. In the current context, we assume code matrices $\boldsymbol{C}_{1,1}$ and $\boldsymbol{C}_{1,2}$ are known. This indicates that both the spreading sequence and delay for this user is given. Following similar steps in the previous section and using a definition similar to (13), we have a new error matrix

$$\boldsymbol{E}_1(n) = \boldsymbol{C}_{1,1}\boldsymbol{D}_1\boldsymbol{C}_{1,1}^H + \boldsymbol{C}_{1,2}\boldsymbol{D}_1\boldsymbol{C}_{1,2}^H + \boldsymbol{R}_i - \boldsymbol{y}(n)\boldsymbol{y}^H(n). \tag{41}$$

Defining the index of this error at time n in the similar form as (16), we obtain

$$J_1(n) = \{\boldsymbol{Q}_1\boldsymbol{d}_1 + \text{vec}(\boldsymbol{R}_i) - \text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}^H \cdot \{\boldsymbol{Q}_1\boldsymbol{d}_1 + \text{vec}(\boldsymbol{R}_i) - \text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\} \tag{42}$$

where

$$\boldsymbol{Q}_1 = \boldsymbol{C}_{1,1}^* \otimes \boldsymbol{C}_{1,1} + \boldsymbol{C}_{1,2}^* \otimes \boldsymbol{C}_{1,2},\ \boldsymbol{d}_1 = \text{vec}(\boldsymbol{D}_1).$$

First, we investigate our batch approach and construct the cost function

$$J_1 = \frac{1}{N}\sum_{n=1}^{N} J_1(n) \tag{43}$$

which after substituting (42) is

$$J_1 = \frac{1}{N}\sum_{n=1}^{N}\{\boldsymbol{Q}_1\boldsymbol{d}_1 + \text{vec}(\boldsymbol{R}_i) - \text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}^H \cdot \{\boldsymbol{Q}_1\boldsymbol{d}_1 + \text{vec}(\boldsymbol{R}_i) - \text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}. \tag{44}$$

This is a quadratic function of both vec$(\boldsymbol{R}_i)$ and $\boldsymbol{d}_1$. If we take the derivative with respect to vec$(\boldsymbol{R}_i)$ first and set it equal to zero, we can obtain an estimate for vec$(\boldsymbol{R}_i)$. Under this estimate, the derivative with respect to $\boldsymbol{d}_1$ can be easily shown to be

$$\nabla_{\boldsymbol{d}_1^H} J = \boldsymbol{F}_N \boldsymbol{d}_1 - \boldsymbol{f}_N \tag{45}$$

where

$$\begin{aligned}\boldsymbol{F}_N &= \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{Q}_1^H\boldsymbol{Q}_1) - \frac{1}{N^2}\left(\sum_{n=1}^{N}\boldsymbol{Q}_1^H\right)\left(\sum_{n=1}^{N}\boldsymbol{Q}_1\right)\\ \boldsymbol{f}_N &= \frac{1}{N}\sum_{n=1}^{N}\{\boldsymbol{Q}_1^H \text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}\\ &\quad - \frac{1}{N^2}\left(\sum_{n=1}^{N}\boldsymbol{Q}_1^H\right)\sum_{n=1}^{N}\text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)].\end{aligned}$$

By setting (45) equal to zero, we can obtain our estimate $\hat{\boldsymbol{d}}_1$ for $\boldsymbol{d}_1$

$$\hat{\boldsymbol{d}}_1 = \boldsymbol{F}_N^{-1}\boldsymbol{f}_N. \tag{46}$$

Once $\hat{\boldsymbol{d}}_1$ is obtained, $\hat{\boldsymbol{D}}_1$ can be reconstructed by the reverse of the "vec" operation.

Following the same procedure as in Appendix B and considering (68) and (77) therein, we can obtain the asymptotic result as $N \to \infty$

$$\boldsymbol{F}_N \stackrel{w.p.1}{\longrightarrow} \boldsymbol{F}, \qquad \boldsymbol{f}_N \stackrel{w.p.1}{\longrightarrow} \boldsymbol{F}\boldsymbol{d}_1 \tag{47}$$

where

$$\boldsymbol{F} = \boldsymbol{\Delta}_{1,1} - \sigma_c^4 \boldsymbol{B}^T\boldsymbol{B} \tag{48}$$

$$\begin{aligned}\boldsymbol{B} = \sum_{l=0}^{P-1}\Big[&\left(\boldsymbol{X}^{l+\delta_1}\boldsymbol{M}\right)\otimes\left(\boldsymbol{X}^{l+\delta_1}\boldsymbol{M}\right)\\ &+\left(\tilde{\boldsymbol{X}}^{P-\delta_1}\boldsymbol{X}^l\boldsymbol{M}\right)\otimes\left(\tilde{\boldsymbol{X}}^{P-\delta_1}\boldsymbol{X}^l\boldsymbol{M}\right)\Big]\end{aligned} \tag{49}$$

and $\boldsymbol{\Delta}_{1,1}$ is given in Appendix B. Therefore, if $\boldsymbol{F}$ is nonsingular, then the solution of (46) will asymptotically converge to the true vector $\boldsymbol{d}_1$. The rank of this matrix can also be checked by computer. It is also found in our test that for all possible $P < 128$, $q < 20$, and $\delta_1 < P$, this square matrix has full rank.

This algorithm provides a simpler procedure to blindly estimate the channel for the user of interest with less computational complexity [about $(q+1)^6$] since we only need to solve $(q+1)^2$ equations for $(q+1)^2$ unknown parameters in $\hat{\boldsymbol{d}}_1$.

We may wonder if a similar adaptive version can be obtained. Since we have different structure of the model ($\boldsymbol{R}_i$ instead of σ_v^2 in the output correlation matrix with more parameters), the cost function $J_1(n)$ at time n is overparameterized by $\boldsymbol{D}_1$ and $\boldsymbol{R}_i$. Therefore, the instantaneous approximation of the gradient cannot be used directly. Instead, we keep L ($L \geq 2$) data vectors from time $n - L + 1$ up to n and approximate this gradient at time n as [see (45)]

$$\begin{aligned}\nabla_{\boldsymbol{d}_1^H} J_1(n) = \Bigg\{&\frac{1}{L}\sum_{l=n-L+1}^{n}[\boldsymbol{Q}_1^H(l)\boldsymbol{Q}_1(l)] - \frac{1}{L^2}\\ &\cdot\left[\sum_{l=n-L+1}^{n}\boldsymbol{Q}_1^H(l)\right]\left[\sum_{l=n-L+1}^{n}\boldsymbol{Q}_1(l)\right]\Bigg\}\boldsymbol{d}_1^{(n)}\\ &-\frac{1}{L}\sum_{l=n-L+1}^{n}\{\boldsymbol{Q}_1^H(l)\text{vec}[\boldsymbol{y}(l)\boldsymbol{y}^H(l)]\} + \frac{1}{L^2}\\ &\cdot\left[\sum_{l=n-L+1}^{n}\boldsymbol{Q}_1^H(l)\right]\sum_{l=1}^{L}\text{vec}\left[\boldsymbol{y}(l)\boldsymbol{y}^H(l)\right]\end{aligned} \tag{50}$$

where we add the time index back for $\boldsymbol{Q}_1$ to avoid ambiguity. According to (50), $\boldsymbol{d}_1$ can be updated as

$$\boldsymbol{d}_1^{(n+1)} = \boldsymbol{d}_1^{(n)} - \mu_1 \nabla_{\boldsymbol{d}_1^H} J_1(n) \tag{51}$$

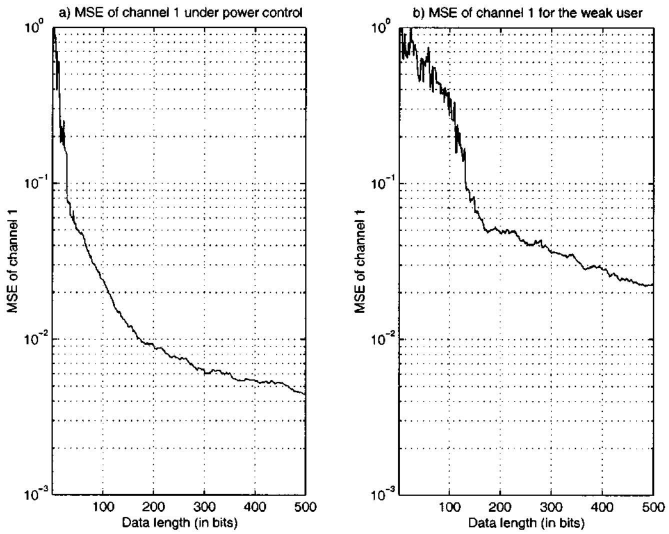
where μ_1 is the step size. In this approach, the computational complexity is about $K^2L^2(q+1)^2$. The factor L will affect both this complexity and convergence rate of this algorithm, and therefore, there is a compromise between them. The larger L is, the faster the algorithm converges, but the more computations it requires.

VI. Simulations

In this section, we will show some computer simulation results for our channel estimation algorithms both for the multiuser (in Section III) and single-user (in Section V) case. For simplicity, notations MU (multiuser) and SU (single-user) will be used to represent these two methods. The performance of both batch and adaptive algorithms is investigated.

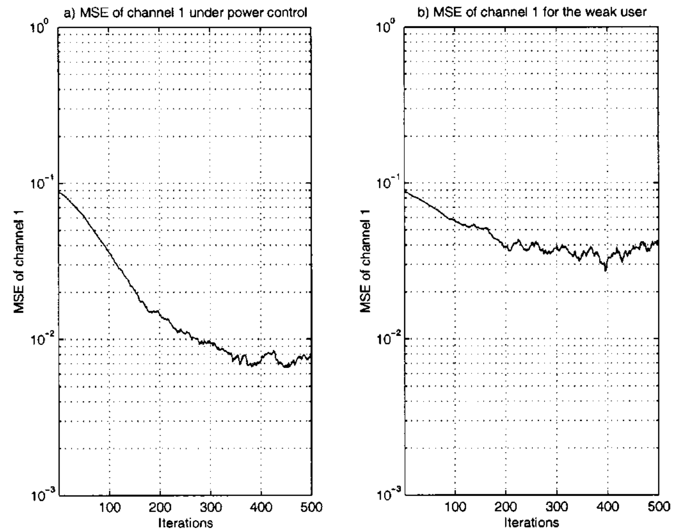
In our experiment, a DS-CDMA system is simulated. All users' long spreading sequences and transmitted bits are assumed i.i.d. taking values from $\{+1, -1\}$. The spreading factor for each bit is set to be 16 (as in [24]) for every user. The maximum communication channel order for all users is 3 ($q = 3$). The 15 dB AWGN noise is added to the system.[3] The channel coefficients for all users are randomly generated and kept constant. Since our estimate for channel vector $\boldsymbol{g}_j$ is obtained from SVD operation, a complex scalar ambiguity exists. For comparison, this ambiguity should be eliminated. To consider the phase ambiguity first, we calibrate the phase for our estimate $\hat{g}_j$ according to the phase of a particular element in $\boldsymbol{g}_j$. For example, if we use its first element $g_j(0)$ as a phase reference, denote its phase by $\phi_{j,0}$, and the phase of $\hat{g}_{j,0}$ (the first element in $\hat{\boldsymbol{g}}_j$) by $\hat{\phi}_{j,0}$; then, our estimate becomes $\hat{\boldsymbol{g}}_j e^{i\phi_{j,0} - i\hat{\phi}_{j,0}}$. We will not introduce a new notation for this phase-calibrated estimate later but will still use $\hat{\boldsymbol{g}}_j$ to simplify the presentation. Second, in order to avoid the norm ambiguity, we choose the following mean square error (MSE) as the performance measure: $(1/K_1)\sum_{k=1}^{K_1}||\hat{\boldsymbol{g}}_j^{(k)} - (\boldsymbol{g}_j/||\boldsymbol{g}_j||)||^2$, where K_1 is the number of Monte Carlo runs, which is set to be 50 in our experiments.

[3]Fifteen dB is the bit SNR and not the chip SNR.



(a) (b)

Fig. 2. MSE of channel 1 (MU batch method).



(a) (b)

Fig. 3. MSE of channel 1 (MU adaptive method).

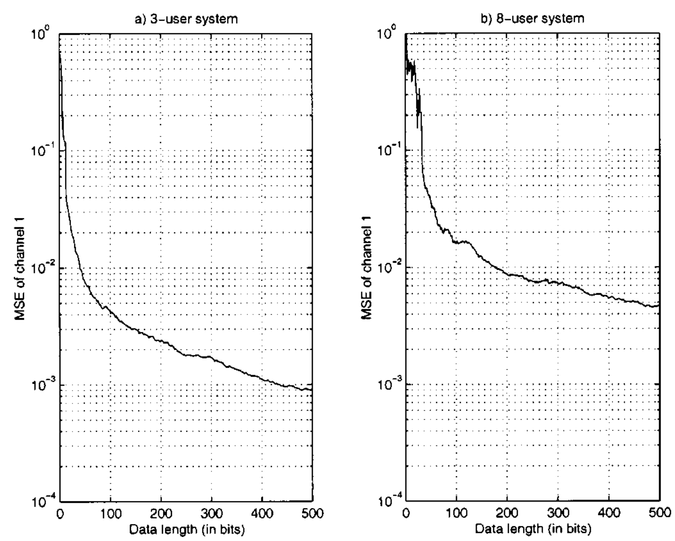
We first test our MU batch method with eight users in the system ($M = 8$). Each user transmitted 500 bits. All users are assigned equal power. The typical result of channel estimation error (e.g., channel 1) is presented in Fig. 2(a). It can be seen from this figure that the MSE for channel estimates reaches 5×10^{-3} after 500 bits are transmitted. In order to see the near-far effect on the performance of our method, we let user 1 to be 5 dB weaker in power than all other equally powered users and repeat the experiment. The result for this weak user is plotted in Fig. 2(b). Compared with Fig. 2(a), a penalty line estimation occurs due to the imperfect power control. For example, the error becomes 2×10^{-2} after 500 bit periods.

We also test our adaptive method under those two different power assignments as above. In implementing the algorithm, each bit corresponds to one iteration, the step size is chosen to be 3×10^{-5}, and initial values for all channels are set to be $[1, 1, 1, 1]^T$. The results for channel 1 are depicted in Fig. 3(a) and (b), corresponding to those two cases, respectively. Under power control, the MSE for channel 1 achieves about an 8×10^{-3} level after 500 iterations, according to Fig. 3(a), whereas it increases to 4×10^{-2} when the power for user 1 is reduced. Similarly, performance loss is observed if there is a near-far problem. It is also evident that compared with the batch method, the adaptive algorithm converges more slowly and to a higher MSE level due to the excess error common in all LMS-type algorithms (see [12]).

We also study the performance of our MU batch method in some extreme situations. Three typical cases are considered here

1) Some users experience the same communication channel.
2) Maximum channel order is overestimated.
3) Maximum channel order is underestimated.

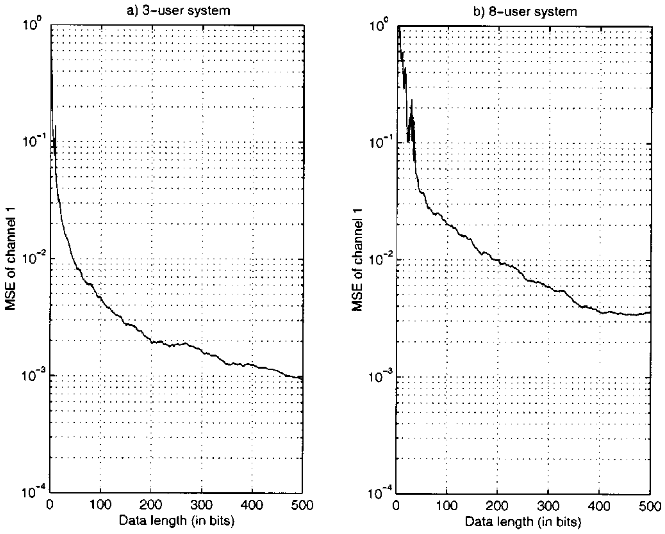
In each case, MSE's of channel estimates are obtained both for a three-user system and for an eight-user system to reflect the influence from different system loads. Those results are shown in Figs. 4–6, respectively. Fig. 4 uses the same channel for user 1 and 2 ($\boldsymbol{g}_1 = \boldsymbol{g}_2$), which is different from all others. From this figure, we can conclude that low error level can still be



(a) (b)

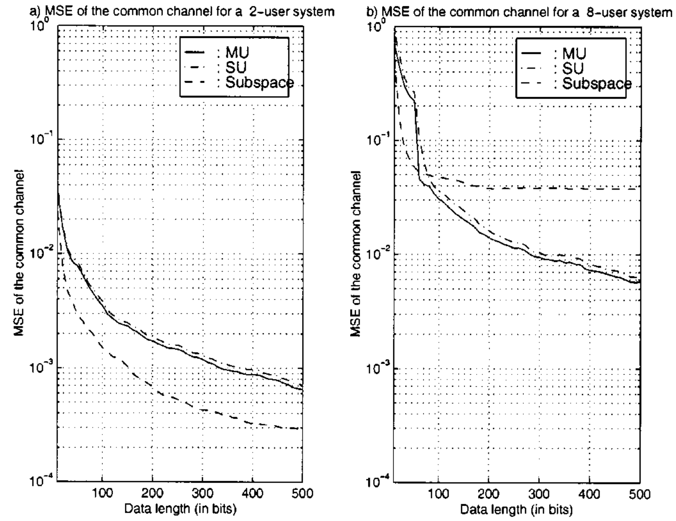
Fig. 4. MSE's of channel 1 for a three-user system and eight-user system with $\boldsymbol{g}_1 = \boldsymbol{g}_2$.

achieved in case 1. Moreover, from Fig. 4(a) and (b), we can see that the channel estimation error increases with the number of users. For Fig. 5, the true channel order is 2 for all users, whereas the maximum channel order is assumed to be 3, which is overestimated. The convergence level at about 10^{-3} can be observed from Fig. 5(a) after 500 input bits, and 3×10^{-3} from Fig. 5(b). In case 3, we underestimate the maximum channel order by assigning channel for each user with order 3 while setting the maximum order to be 2 in the algorithm. Therefore, the fourth tap coefficient is not estimated. The MSE of channel 1 is plotted in Fig. 6(a) for a three-user system and in Fig. 6(b) for an eight-user system. This time the error levels are increased compared with Fig. 5 due to the omission of the fourth element in the channel vector. This indicates that the maximum channel order should be large enough to cover all the channel length. In some cases, this can be *a priori* known. However, it is true that



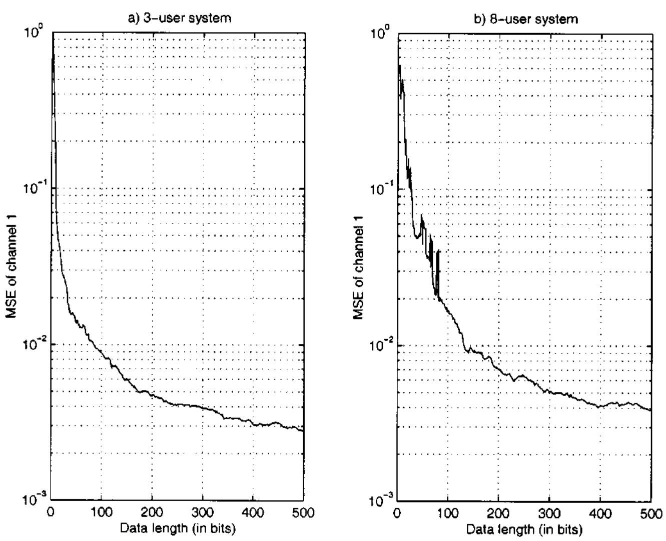
(a) (b)

Fig. 5. MSE of channel 1 for a three-user system and eight-user system with channel order over-estimation.



(a) (b)

Fig. 7. Comparison of the proposed MU and SU batch methods with subspace method for a two-user and an eight-user system.



(a) (b)

Fig. 6. MSE of channel 1 for a three-user system and eight-user system with channel order underestimation.

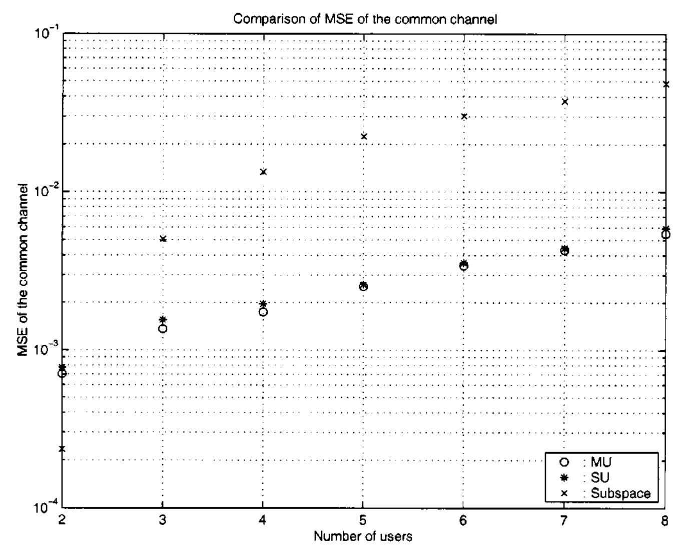


Fig. 8. Comparison of the proposed MU and SU batch methods with subspace method after 500 bits are transmitted.

more unknowns could be introduced, and thus, the higher computational cost is expected.

Our next experiment is to compare the proposed MU batch method with the SU one and the recently presented subspace method [31] for systems with different loads. We assume user 1 is the desired user and that each user has 500 binary bits to transmit through a common FIR channel of order 3. We implemented the subspace approach in [31] by assigning user 1 in group 1 and all other users in group 2 so that the matrix $\tilde{C}_{m,1}$ would be left invertible, which is required in that context (see [31] for details). According to this grouping method, it is fair to compare it with our SU method because both of them treat other users as interfering users. The channel estimation errors for a two-user system are compared in Fig. 7(a) and for an eight-user system in Fig. 7(b). Solid lines represent the results of the MU method, dash-dotted lines for the SU method, and dashed lines for the subspace method. From Fig. 7, it can be observed first that the MU algorithm performs better than the SU one. This difference is more evident, especially for a system with heavy load as in Fig. 7(b). This, of course, comes at the expense of increased computational complexity. Second, the subspace method has better performance than the proposed SU and MU algorithms when the system has only a few users according to Fig. 7(a), whereas it becomes inferior under heavy load as it converges to a higher error level (4×10^{-2}). However, the MSE of the proposed methods can converge to 6×10^{-3} after 500 input bits. A detailed comparison of the MSE's with respect to different number of users is shown in Fig. 8. The circles are the results for the proposed MU method and the stars for the proposed SU method, whereas we use "×" for the subspace method. The similar conclusion can also be easily made from Fig. 7.

Our methods are derived under the assumption of time-invariant channels. This can be hardly met in some communi-

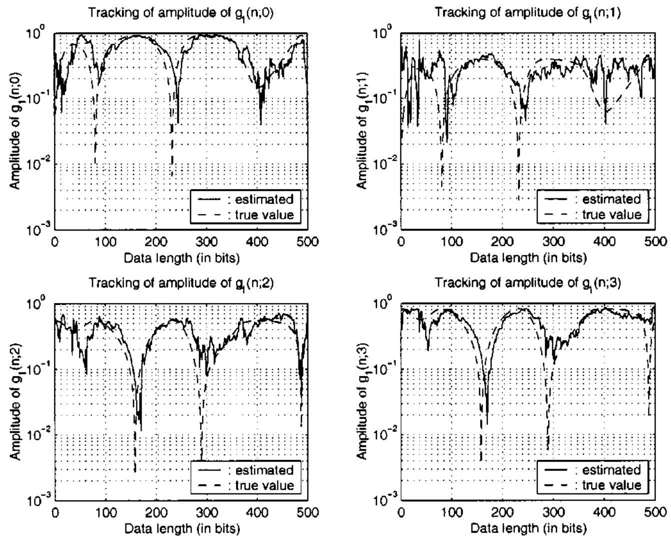


Fig. 9. Tracking of the amplitude of fading channel coefficients for a three-user system.

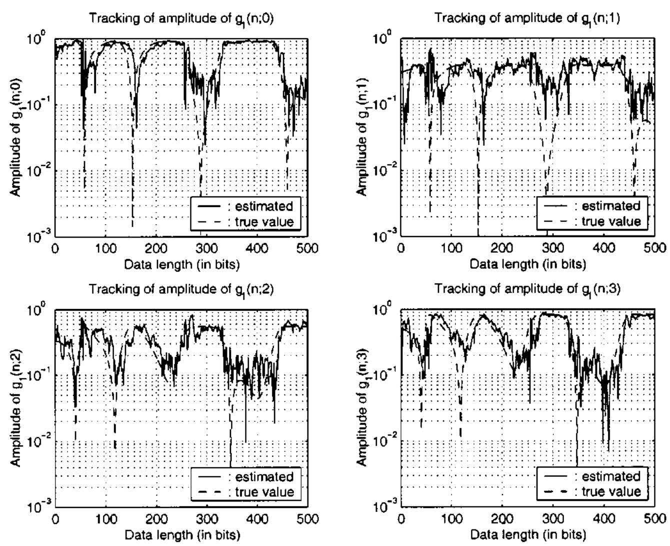


Fig. 10. Tracking of the amplitude of fading channel coefficients for an eight-user system.

cation environments. We may wonder if the MU adaptive algorithm is applicable for tracking time-varying channel taps. We will test it for a Rayleigh fading communication system. However, examples shown next are only intended to illustrate the use of the proposed method and should not be regarded as a detailed simulations study. Similar to the previous experiments, both three-user and eight-user systems are considered. Each user has information to transmit through a two-ray independent Rayleigh fading channel. The maximum order of all different channels is equal to 3. The carrier frequency is 900 MHz, and the symbol rate is 9600 b/s. The speeds for different users are randomly chosen in $50 \sim 100$ km/h. The channel coefficients are then generated based on a Rayleigh simulator, e.g., [19]. The initial values for all channel vectors are set to be $[0, 0, 0, 0]^T$. The step size is chosen to be 5×10^{-4}. For such a setup, each channel vector $\boldsymbol{g}_j(n)$ has four elements $g_j(n;0)$, $g_j(n;1)$, $g_j(n;2)$, and $g_j(n;3)$ corresponding to four tap coefficients. To avoid phase ambiguity, we adopt the amplitude of each element as a performance measure for comparison and apply our method to track those four amplitudes. A typical result for channel 1 for a three-user system is plotted in Fig. 9. Dashed lines represent the true values after normalization, whereas solid lines are their estimates. The deep fading can be seen from this figure, and its waveform can be tracked based on the proposed method. However, a larger deviation exists within 100 bit periods. With increasingly more received data, the tracking performance is improved for some of the four elements. When there are more users active in the system, the reliable estimate for time-varying channel coefficients are still obtained as indicated by Fig. 10, although the tracking performance degrades.

VII. Conclusions

Our experimental results and performance analysis demonstrate the applicability of correlation techniques in estimating channel parameters for long code multiuser CDMA communication systems. Furthermore, under mild conditions, these estimators asymptotically converge to the true channels and are not sensitive to order overestimation. They also exhibit good performance, even when the system is heavily loaded. For reduced computational complexity, alternatives of single-user estimators with both batch and adaptive algorithms are provided. Comparison results show that the performance of our single-user channel estimation method is inferior than, but very close to, the proposed multiuser one. Furthermore, our methods are suitable even for heavily loaded systems, in which case, subspace methods show unsatisfactory performance.

Appendix A
Derivation of (29)

In order to solve for $\boldsymbol{d}$ and x, we take the derivative of (27) with respect to x^* and $\boldsymbol{d}^H$, respectively

$$\nabla_{x^*} J = Kx + \frac{1}{N}\sum_{n=1}^{N} \cdot \left\{\mathrm{vec}^T(\boldsymbol{I})\boldsymbol{Q}\boldsymbol{d} - \mathrm{vec}^T(\boldsymbol{I})\mathrm{vec}\left[\boldsymbol{y}(n)\boldsymbol{y}^H(n)\right]\right\} \quad (52)$$

$$\nabla_{\boldsymbol{d}^H} J = \frac{1}{N}\sum_{n=1}^{N}\left\{\boldsymbol{Q}^H\boldsymbol{Q}\boldsymbol{d} + \boldsymbol{Q}^H\mathrm{vec}(\boldsymbol{I})x - \boldsymbol{Q}^H\mathrm{vec}\left[\boldsymbol{y}(n)\boldsymbol{y}^H(n)\right]\right\}. \quad (53)$$

At the equilibrium point ($\hat{x}_N$, $\hat{\boldsymbol{d}}_N$) of our cost function J, these derivatives are equal to zero. By setting (52) to zero, $\hat{x}_N$ is solved first as

$$\hat{x}_N = \frac{1}{KN}\sum_{n=1}^{N} \cdot \left\{\mathrm{vec}^T(\boldsymbol{I})\mathrm{vec}\left[\boldsymbol{y}(n)\boldsymbol{y}^H(n)\right] - \mathrm{vec}^T(\boldsymbol{I})\boldsymbol{Q}\hat{\boldsymbol{d}}_N\right\}. \quad (54)$$

Equation (54) can be simplified after considering (15) and (22). First, using (15), we have

$$\mathrm{vec}^T(\boldsymbol{I})\mathrm{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)] = \mathrm{tr}\left[\boldsymbol{y}(n)\boldsymbol{y}^H(n)\right] = \boldsymbol{y}^H(n)\boldsymbol{y}(n).$$

Observe that $\text{vec}^T(\boldsymbol{I})\boldsymbol{Q} = [\boldsymbol{Q}^H\text{vec}(\boldsymbol{I})]^H$. By using (22), the jth subvector $\boldsymbol{Q}_j^H\text{vec}(\boldsymbol{I})$ in $\boldsymbol{Q}^H\text{vec}(\boldsymbol{I})$ can be written as $\boldsymbol{Q}_j^H\text{vec}(\boldsymbol{I}) = \text{vec}(\boldsymbol{H}_j)$. Therefore, $\boldsymbol{Q}^H\text{vec}(\boldsymbol{I}) = \boldsymbol{b}(n)$. Then, (54) becomes

$$\hat{x}_N = \frac{1}{KN}\sum_{n=1}^{N}\left[\boldsymbol{y}^H(n)\boldsymbol{y}(n) - \boldsymbol{b}^H(n)\hat{\boldsymbol{d}}_N\right]. \quad (55)$$

Substituting (55) in (53) and setting it equal to zero, (29) follows. □

Appendix B
Derivation of (38)

We start our derivation from (29), where our estimator is obtained. Both the convergence properties of $\boldsymbol{T}_N$ and $\boldsymbol{t}_N$ in the equation will be investigated. Let us first focus on the first term of $\boldsymbol{T}_N$ and $\boldsymbol{t}_N$ [cf. (30) and (31)] without the summation, that is, $\boldsymbol{A}(n) = \boldsymbol{Q}^H\boldsymbol{Q}$ and $\boldsymbol{a}(n) = \boldsymbol{Q}^H\text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]$. From the expression of $\boldsymbol{Q}$ in (25), it can be seen that each element in matrix $\boldsymbol{A}(n)$ is the fourth-order product of the spreading codes. Similarly, by recalling (10), $\boldsymbol{a}(n)$ also contains products of the codes up to the fourth order and products of the codes with Gaussian noise. Under our assumptions, it can be easily shown that $\boldsymbol{A}(n)$ and $\boldsymbol{a}(n)$ are stationary with finite and absolutely summable cumulants of any order. As $N \to \infty$, their sample averages therefore strongly converge to their ensemble averages, e.g., [2], that is

$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{A}(n) \xrightarrow{w.p.1} E\{\boldsymbol{A}(n)\}, \quad \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{a}(n) \xrightarrow{w.p.1} E\{\boldsymbol{a}(n)\}.$$

Similarly, we obtain[4]

$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{b}(n) \xrightarrow{w.p.1} E\{\boldsymbol{b}(n)\}$$
$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{y}^H(n)\boldsymbol{y}(n) \xrightarrow{w.p.1} E\{\boldsymbol{y}^H(n)\boldsymbol{y}(n)\}.$$

Putting everything together, we obtain the limits

$$\lim_{N\to\infty}\boldsymbol{T}_N \xrightarrow{w.p.1} \boldsymbol{T},\ \boldsymbol{T} = E\{\boldsymbol{Q}^H\boldsymbol{Q}\} - \frac{1}{K}E\{\boldsymbol{b}(n)\}E\left\{\boldsymbol{b}^H(n)\right\} \quad (56)$$

$$\lim_{N\to\infty}\boldsymbol{t}_N \xrightarrow{w.p.1} \boldsymbol{t},\ \boldsymbol{t} = E\{\boldsymbol{Q}^H\text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\} - \frac{1}{K}E\{\boldsymbol{y}^H(n)\boldsymbol{y}(n)\}E\{\boldsymbol{b}(n)\} \quad (57)$$

where we have used the fact that the product of sample averages of two random variables with finite cumulants strongly converge to the product of the limits. If we define $\hat{\boldsymbol{d}}_\infty = \lim_{N\to\infty}\hat{\boldsymbol{d}}_N$, then from (29), our estimator satisfies

$$\boldsymbol{T}\hat{\boldsymbol{d}}_\infty = \boldsymbol{t}. \quad (58)$$

[4]Do not confuse $E\{\boldsymbol{y}^H(n)\boldsymbol{y}(n)\}$ with the correlation defined in (11). In the current case, $\boldsymbol{y}(n)$ is not conditioned on any code $\boldsymbol{C}_j(n)$ and is therefore stationary.

To evaluate $\boldsymbol{t}$ given by (57), the output $\boldsymbol{y}(n)$ is required. According to our input/output model (10), its first term $E\{\boldsymbol{Q}^H\text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\}$ can be evaluated as

$$E\{\boldsymbol{Q}^H\text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\} = E\{\boldsymbol{Q}^H\boldsymbol{Q}\}\boldsymbol{d} + \sigma_v^2E\{\boldsymbol{Q}^H\text{vec}(\boldsymbol{I})\} \quad (59)$$

whereas all other terms vanish due to the zero-mean, i.i.d. properties of the inputs and AWGN. Since $\boldsymbol{Q}^H\text{vec}(\boldsymbol{I}) = \boldsymbol{b}(n)$, (59) becomes

$$E\{\boldsymbol{Q}^H\text{vec}[\boldsymbol{y}(n)\boldsymbol{y}^H(n)]\} = E\{\boldsymbol{Q}^H\boldsymbol{Q}\}\boldsymbol{d} + \sigma_v^2E\{\boldsymbol{b}(n)\}. \quad (60)$$

For $E\{\boldsymbol{y}^H(n)\boldsymbol{y}(n)\}$ in the second term, it is not hard to obtain

$$\begin{aligned} &E\{\boldsymbol{y}^H(n)\boldsymbol{y}(n)\} \\ &= E\{\boldsymbol{v}^H(n)\boldsymbol{v}(n)\} + E\left\{\sum_{j=1}^{M}\text{tr}\left[(\boldsymbol{C}_{j,1}^H\boldsymbol{C}_{j,1} + \boldsymbol{C}_{j,2}^H\boldsymbol{C}_{j,2})\boldsymbol{D}_j\right]\right\} \\ &= \sigma_v^2K + E\{\boldsymbol{b}^H(n)\}\boldsymbol{d}. \end{aligned} \quad (61)$$

Therefore, $\boldsymbol{t}$ can be computed according to (60), (61), and (57)

$$\boldsymbol{t} = \boldsymbol{T}\boldsymbol{d}. \quad (62)$$

Substituting (62) in (58), we obtain (38). Next, we will derive a closed-form expression for the constant matrix $\boldsymbol{T}$.

We know from (56) that $\boldsymbol{T}$ depends on $E\{\boldsymbol{Q}^H(n)\boldsymbol{Q}(n)\}$ and $E\{\boldsymbol{b}(n)\}$. Let us first focus on $E\{\boldsymbol{Q}^H(n)\boldsymbol{Q}(n)\}$. Its (k, j)th submatrix is $E\{\boldsymbol{Q}_k^H(n)\boldsymbol{Q}_j(n)\}$ for $k, j = 1, \cdots, M$. Let us denote them by

$$\begin{aligned} \boldsymbol{\Delta} &= E\{\boldsymbol{Q}^H(n)\boldsymbol{Q}(n)\} \\ &= \begin{bmatrix} \boldsymbol{\Delta}_{1,1} & \cdots & \boldsymbol{\Delta}_{1,M} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Delta}_{M,1} & \cdots & \boldsymbol{\Delta}_{M,M} \end{bmatrix} \\ \boldsymbol{\Delta}_{k,j} &= E\left\{\boldsymbol{Q}_k^H(n)\boldsymbol{Q}_j(n)\right\}. \end{aligned} \quad (63)$$

Matrix $\boldsymbol{Q}_j(n)$ is related to code matrices by its definition (25). Those code matrices stem from $\boldsymbol{C}_j$ by shift operation, which is determined by delay δ_j [see (8) and (9)]. Since each subdiagonal of $\boldsymbol{C}_j$ has the same element, we can write $\boldsymbol{C}_j$ as

$$\boldsymbol{C}_j(n) = \sum_{l=0}^{P-1}c_{j,n}(l)\boldsymbol{X}^l\boldsymbol{M} \quad (64)$$

where $\boldsymbol{X}$ is a Jordan matrix of dimension $K \times K$, and $\boldsymbol{M}$ is a $K \times (q+1)$ matrix

$$\boldsymbol{X} = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & 1 & 0 \end{bmatrix}, \qquad \boldsymbol{M} = \begin{bmatrix} \boldsymbol{I}_{q+1} \\ \boldsymbol{0} \end{bmatrix}$$
$$\tilde{\boldsymbol{X}} = \boldsymbol{X}^T, \qquad \tilde{\boldsymbol{M}} = \boldsymbol{M}^T. \quad (65)$$

For notational convenience, $\boldsymbol{X}^T$ and $\boldsymbol{M}^T$ are denoted by $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{M}}$ in (65), respectively. Then, $\boldsymbol{C}_{j,1}(n)$, $\boldsymbol{C}_{j,2}(n-1)$ can be obtained from $\boldsymbol{C}_j(n)$ and $\boldsymbol{C}_j(n-1)$ by the shift operation[5]

$$\begin{aligned}\boldsymbol{C}_{j,1}(n) &= \boldsymbol{X}^{\delta_j}\boldsymbol{C}_j(n)\\ \boldsymbol{C}_{j,2}(n-1) &= \tilde{\boldsymbol{X}}^{P-\delta_j}\boldsymbol{C}_j(n-1).\end{aligned} \tag{66}$$

Substituting (64) in (66), we obtain

$$\begin{aligned}\boldsymbol{C}_{j,1}(n) &= \sum_{l=0}^{P-1} c_{j,n}(l)\boldsymbol{X}^{l+\delta_j}\boldsymbol{M}\\ \boldsymbol{C}_{j,2}(n-1) &= \sum_{l=0}^{P-1} c_{j,n-1}(l)\tilde{\boldsymbol{X}}^{P-\delta_j}\boldsymbol{X}^{l}\boldsymbol{M}.\end{aligned} \tag{67}$$

Then, from (25), $\boldsymbol{Q}_j$ becomes

$$\begin{aligned}\boldsymbol{Q}_j = &\sum_{l_1,l_2=0}^{P-1} c_{j,n}^*(l_1)c_{j,n}(l_2)(\boldsymbol{X}^{l_1+\delta_j}\boldsymbol{M})\otimes(\boldsymbol{X}^{l_2+\delta_j}\boldsymbol{M})\\ &+\sum_{l_1,l_2=0}^{P-1} c_{j,n-1}^*(l_1)c_{j,n-1}(l_2)(\tilde{\boldsymbol{X}}^{P-\delta_j}\boldsymbol{X}^{l_1}\boldsymbol{M})\\ &\otimes(\tilde{\boldsymbol{X}}^{P-\delta_j}\boldsymbol{X}^{l_2}\boldsymbol{M}).\end{aligned} \tag{68}$$

Based on the properties of the Kronecker product (see [10, ch. 12])

$$\begin{aligned}(\boldsymbol{X}_1\otimes\boldsymbol{X}_2)^T &= \boldsymbol{X}_1^T\otimes\boldsymbol{X}_2^T\\ (\boldsymbol{X}_1\boldsymbol{X}_2)\otimes(\boldsymbol{X}_3\boldsymbol{X}_4) &= (\boldsymbol{X}_1\otimes\boldsymbol{X}_3)(\boldsymbol{X}_2\otimes\boldsymbol{X}_4)\end{aligned}$$

$\boldsymbol{Q}_k^H$ becomes

$$\begin{aligned}\boldsymbol{Q}_k^H = &\sum_{l_1,l_2=0}^{P-1} c_{k,n}(l_1)c_{k,n}^*(l_2)(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1+\delta_k})\otimes(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_2+\delta_k})\\ &+\sum_{l_1,l_2=0}^{P-1} c_{k,n-1}(l_1)c_{k,n-1}^*(l_2)(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1}\boldsymbol{X}^{P-\delta_k})\\ &\otimes(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_2}\boldsymbol{X}^{P-\delta_k}).\end{aligned} \tag{69}$$

Therefore, $\boldsymbol{\Delta}_{k,j}$ in (63) can be easily expressed by

$$\boldsymbol{\Delta}_{k,j} = \boldsymbol{S}_1+\boldsymbol{S}_2+\boldsymbol{S}_3+\boldsymbol{S}_4 \tag{70}$$

where

$$\begin{aligned}\boldsymbol{S}_1 = &\sum_{l_1,l_2,l_3,l_4=0}^{P-1} E\{c_{k,n}(l_1)c_{k,n}^*(l_2)c_{j,n}^*(l_3)c_{j,n}(l_4)\}\\ &\cdot\boldsymbol{U}_1(l_1,l_2,l_3,l_4,k,j)\\ \boldsymbol{S}_2 = &\sum_{l_1,l_2,l_3,l_4=0}^{P-1} E\{c_{k,n}(l_1)c_{k,n}^*(l_2)c_{j,n-1}^*(l_3)c_{j,n-1}(l_4)\}\\ &\cdot\boldsymbol{U}_2(l_1,l_2,l_3,l_4,k,j)\\ \boldsymbol{S}_3 = &\sum_{l_1,l_2,l_3,l_4=0}^{P-1} E\{c_{k,n-1}(l_1)c_{k,n-1}^*(l_2)c_{j,n}^*(l_3)c_{j,n}(l_4)\}\\ &\cdot\boldsymbol{U}_3(l_1,l_2,l_3,l_4,k,j)\\ \boldsymbol{S}_4 = &\sum_{l_1,l_2,l_3,l_4=0}^{P-1} E\{c_{k,n-1}(l_1)c_{k,n-1}^*(l_2)c_{j,n-1}^*(l_3)c_{j,n-1}(l_4)\}\\ &\cdot\boldsymbol{U}_4(l_1,l_2,l_3,l_4,k,j)\end{aligned} \tag{71}$$

and

$$\begin{aligned}\boldsymbol{U}_1(l_1,l_2,l_3,l_4,k,j) &= (\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1+\delta_k}\boldsymbol{X}^{l_3+\delta_j}\boldsymbol{M})\\ &\quad\otimes(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_2+\delta_k}\boldsymbol{X}^{l_4+\delta_j}\boldsymbol{M})\\ \boldsymbol{U}_2(l_1,l_2,l_3,l_4,k,j) &= (\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{P+l_1+\delta_k-\delta_j}\boldsymbol{X}^{l_3}\boldsymbol{M})\\ &\quad\otimes(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{P+l_2+\delta_k-\delta_j}\boldsymbol{X}^{l_4}\boldsymbol{M})\\ \boldsymbol{U}_3(l_1,l_2,l_3,l_4,k,j) &= (\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1}\boldsymbol{X}^{P+l_3+\delta_j-\delta_k}\boldsymbol{M})\\ &\quad\otimes(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_2}\boldsymbol{X}^{P+l_4+\delta_j-\delta_k}\boldsymbol{M})\\ \boldsymbol{U}_4(l_1,l_2,l_3,l_4,k,j) &= (\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1}\boldsymbol{X}^{P-\delta_k}\tilde{\boldsymbol{X}}^{P-\delta_j}\boldsymbol{X}^{l_3}\boldsymbol{M})\\ &\quad\otimes(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_2}\boldsymbol{X}^{P-\delta_k}\tilde{\boldsymbol{X}}^{P-\delta_j}\boldsymbol{X}^{l_4}\boldsymbol{M}).\end{aligned} \tag{72}$$

Next, it suffices to discuss those expected values for random codes in $\boldsymbol{S}_1\sim\boldsymbol{S}_4$ in different cases.

Case 1—$k\neq j$: In this case, $\boldsymbol{\Delta}_{k,j}$ is an off-diagonal subblock of matrix $\boldsymbol{\Delta}$. Under our i.i.d. assumptions on the random codes, it can be shown for $k\neq j$ that

$$\begin{aligned}\boldsymbol{\Delta}_{k,j} = &\sigma_c^4\sum_{l_1,l_2=0}^{P-1}[\boldsymbol{U}_1(l_1,l_1,l_2,l_2,k,j)\\ &+\boldsymbol{U}_2(l_1,l_1,l_2,l_2,k,j)]\\ &+\sigma_c^4\sum_{l_1,l_2=0}^{P-1}[\boldsymbol{U}_3(l_1,l_1,l_2,l_2,k,j)\\ &+\boldsymbol{U}_4(l_1,l_1,l_2,l_2,k,j)].\end{aligned} \tag{73}$$

Case 2—$k=j$: In this case, $\boldsymbol{\Delta}_{j,j}$ is a diagonal subblock of matrix $\boldsymbol{\Delta}$. Due to i.i.d. codes in time, we can easily obtain $\boldsymbol{S}_2$ and $\boldsymbol{S}_3$ first:

$$\begin{aligned}\boldsymbol{S}_2 &= \sigma_c^4\sum_{l_1,l_2=0}^{P-1}\boldsymbol{U}_2(l_1,l_1,l_2,l_2,j,j)\\ \boldsymbol{S}_3 &= \sigma_c^4\sum_{l_1,l_2=0}^{P-1}\boldsymbol{U}_3(l_1,l_1,l_2,l_2,j,j).\end{aligned} \tag{74}$$

However, there are different cases to be discussed in order to evaluate $\boldsymbol{S}_1$ and $\boldsymbol{S}_4$. Let us first focus on $\boldsymbol{S}_1$. It can be seen from the definition of $\boldsymbol{S}_1$ that when $k=j$, only three cases make contributions to the value of $\boldsymbol{S}_1$.

1) $l_1=l_2=l_3=l_4$ will contribute $m_{4c}\sum_{l=0}^{P-1}\boldsymbol{U}_1(l,l,l,l,j,j)$.
2) $l_1=l_2$, $l_3=l_4$, but $l_1\neq l_3$ will contribute $\sigma_c^4\sum_{l_1,l_3=0,\ l_1\neq l_3}^{P-1}\boldsymbol{U}_1(l_1,l_1,l_3,l_3,j,j)$.
3) $l_1=l_3$, $l_2=l_4$, but $l_1\neq l_2$ will contribute $\sigma_c^4\sum_{l_1,l_2=0,\ l_1\neq l_2}^{P-1}\boldsymbol{U}_1(l_1,l_2,l_1,l_2,j,j)$;

[5]We define $\boldsymbol{X}^0=\tilde{\boldsymbol{X}}^0=\boldsymbol{I}$.

whereas all other cases have no contributions. Therefore, by adding these results up, we can obtain $\boldsymbol{S}_1$ for the case $k = j$

$$\begin{aligned}\boldsymbol{S}_1 = &(m_{4c} - 2\sigma_c^4) \sum_{l=0}^{P-1} \boldsymbol{U}_1(l, l, l, l, j, j) \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \boldsymbol{U}_1(l_1, l_1, l_2, l_2, j, j) \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \boldsymbol{U}_1(l_1, l_2, l_1, l_2, j, j). \end{aligned} \tag{75}$$

Similarly, $\boldsymbol{S}_4$ follows:

$$\begin{aligned}\boldsymbol{S}_4 = &(m_{4c} - 2\sigma_c^4) \sum_{l=0}^{P-1} \boldsymbol{U}_4(l, l, l, l, j, j) \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \boldsymbol{U}_4(l_1, l_1, l_2, l_2, j, j) \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \boldsymbol{U}_4(l_1, l_2, l_1, l_2, j, j). \end{aligned} \tag{76}$$

Combining (74)–(76) and noticing (70), the (j, j)th diagonal subblock $\boldsymbol{\Delta}_{j,j}$ becomes

$$\begin{aligned}\boldsymbol{\Delta}_{j,j} = &(m_{4c} - 2\sigma_c^4) \sum_{l=0}^{P-1} [\boldsymbol{U}_1(l, l, l, l, j, j) \\ &\qquad + \boldsymbol{U}_4(l, l, l, l, j, j)] \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} [\boldsymbol{U}_1(l_1, l_1, l_2, l_2, j, j) \\ &\qquad + \boldsymbol{U}_1(l_1, l_2, l_1, l_2, j, j)] \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} [\boldsymbol{U}_2(l_1, l_1, l_2, l_2, j, j) \\ &\qquad + \boldsymbol{U}_3(l_1, l_1, l_2, l_2, j, j)] \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} [\boldsymbol{U}_4(l_1, l_1, l_2, l_2, j, j) \\ &\qquad + \boldsymbol{U}_4(l_1, l_2, l_1, l_2, j, j)]. \end{aligned} \tag{77}$$

Next, we will evaluate $E\{\boldsymbol{b}(n)\}$ in $\boldsymbol{T}$. Recalling (32), we will compute $E\{\boldsymbol{H}_j\}$ first, which is

$$E\{\boldsymbol{H}_j\} \overset{\text{def}}{=} \boldsymbol{A}_j = E\{\boldsymbol{C}_{j,1}^H \boldsymbol{C}_{j,1} + \boldsymbol{C}_{j,2}^H \boldsymbol{C}_{j,2}\}. \tag{78}$$

According to (67) and the i.i.d. property of the codes, it is not hard to show that

$$\boldsymbol{A}_j = \sigma_c^2 \sum_{l=0}^{P-1} [\tilde{\boldsymbol{M}} \tilde{\boldsymbol{X}}^{l+\delta_j} \boldsymbol{X}^{l+\delta_j} \boldsymbol{M} + \tilde{\boldsymbol{M}} \tilde{\boldsymbol{X}}^{l} \boldsymbol{X}^{P-\delta_j} \tilde{\boldsymbol{X}}^{P-\delta_j} \boldsymbol{X}^{l} \boldsymbol{M}]. \tag{79}$$

Then, $E\{\boldsymbol{b}(n)\}$ can be computed from (79) as

$$E\{\boldsymbol{b}(n)\} \overset{\text{def}}{=} \boldsymbol{b} = \text{vec}[\boldsymbol{A}_1, \cdots, \boldsymbol{A}_M]. \tag{80}$$

Therefore, our constant matrix $\boldsymbol{T}$ can be obtained according to

$$\boldsymbol{T} = \boldsymbol{\Delta} - \frac{1}{K} \boldsymbol{b} \boldsymbol{b}^T \tag{81}$$

based on the following parameters:

- spreading factor P;
- number of users M;
- maximum order of all channels q;
- delays δ_j;
- second- as well as fourth-order moments σ_c^2, m_{4c} of the random spreading codes.

They are all assumed known in this paper. Therefore, $\boldsymbol{T}$ can be precomputed. □

APPENDIX C
DERIVATION OF (39)

If we subtract $\boldsymbol{d}$ on both sides of (37) and take expectation, we obtain

$$\begin{aligned}\triangle \boldsymbol{d}^{(n+1)} = &\left[\boldsymbol{I} - \mu E\{\boldsymbol{Q}^H \boldsymbol{Q}\} + \frac{\mu}{K} E\{\boldsymbol{b}(n) \boldsymbol{b}^H(n)\}\right] \triangle \boldsymbol{d}^{(n)} \\ &+ \frac{\mu}{K} [E\{\boldsymbol{b}(n) \boldsymbol{b}^H(n)\} \boldsymbol{d} - E\{\boldsymbol{y}^H(n) \boldsymbol{y}(n) \boldsymbol{b}(n)\}] \\ &+ \mu E\{\boldsymbol{Q}^H \text{vec}[\boldsymbol{y}(n) \boldsymbol{y}^H(n)]\} - \mu E\{\boldsymbol{Q}^H \boldsymbol{Q}\} \boldsymbol{d}. \end{aligned} \tag{82}$$

According to results (60), (63), and (80) in Appendix B, (82) can be simplified as

$$\begin{aligned}\triangle \boldsymbol{d}^{(n+1)} = &[\boldsymbol{I} - \mu \boldsymbol{\Delta} + \frac{\mu}{K} E\{\boldsymbol{b}(n) \boldsymbol{b}^H(n)\}] \triangle \boldsymbol{d}^{(n)} + \mu \sigma_v^2 \boldsymbol{b} \\ &+ \frac{\mu}{K} [E\{\boldsymbol{b}(n) \boldsymbol{b}^H(n)\} \boldsymbol{d} - E\{\boldsymbol{y}^H(n) \boldsymbol{y}(n) \boldsymbol{b}(n)\}]. \end{aligned} \tag{83}$$

It can be similarly shown as in (61) that

$$E\{\boldsymbol{y}^H(n) \boldsymbol{y}(n) \boldsymbol{b}(n)\} = \sigma_v^2 K \boldsymbol{b} + E\{\boldsymbol{b}(n) \boldsymbol{b}^H(n)\} \boldsymbol{d}. \tag{84}$$

Therefore, (83) becomes

$$\triangle \boldsymbol{d}^{(n+1)} = [\boldsymbol{I} - \mu \boldsymbol{U}] \triangle \boldsymbol{d}^{(n)} \tag{85}$$

where our constant matrix $\boldsymbol{U}$ can be defined as

$$\boldsymbol{U} = \boldsymbol{\Delta} - \frac{1}{K} E\{\boldsymbol{b}(n) \boldsymbol{b}^H(n)\}. \tag{86}$$

Hence, (39) is obtained. Next, we will derive a closed-form expression for matrix $\boldsymbol{U}$ in (86).

We have shown in Appendix B that $\boldsymbol{\Delta}$ is explicitly determined by system parameters $P, M, q, \delta_j, \sigma_c^2$, and m_{4c}. We can similarly evaluate $E\{\boldsymbol{b}(n) \boldsymbol{b}^H(n)\}$ in (86) as in Appendix B. By noticing that $\boldsymbol{b}(n)$ can also be written as

$$\boldsymbol{b}(n) = \begin{bmatrix} \text{vec}[\boldsymbol{H}_1(n)] \\ \vdots \\ \text{vec}[\boldsymbol{H}_M(n)] \end{bmatrix} \tag{87}$$

the (k, j)th subblock of $\boldsymbol{b}(n)\boldsymbol{b}^H(n)$ has a form of vec$[\boldsymbol{H}_k(n)]$vec$^H[\boldsymbol{H}_j(n)]$. Let us define

$$\boldsymbol{\Lambda} = E\{\boldsymbol{b}(n)\boldsymbol{b}^H(n)\} = \begin{bmatrix} \boldsymbol{\Lambda}_{1,1} & \cdots & \boldsymbol{\Lambda}_{1,M} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Lambda}_{M,1} & \cdots & \boldsymbol{\Lambda}_{M,M} \end{bmatrix}$$
$$\boldsymbol{\Lambda}_{k,j} = E\{\text{vec}[\boldsymbol{H}_k(n)]\text{vec}^H[\boldsymbol{H}_j(n)]\} \tag{88}$$

for notational convenience. According to (32) and (67), $\boldsymbol{H}_j(n)$ can be expressed by

$$\begin{aligned}\boldsymbol{H}_j(n) = &\sum_{l_1, l_2=0}^{P-1} c_{j,n}^*(l_1)c_{j,n}(l_2)\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1+\delta_j}\boldsymbol{X}^{l_2+\delta_j}\boldsymbol{M} \\ &+ \sum_{l_1, l_2=0}^{P-1} c_{j,n-1}^*(l_1)c_{j,n-1}(l_2)\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1}\boldsymbol{X}^{P-\delta_j}\tilde{\boldsymbol{X}}^{P-\delta_j} \\ &\cdot \boldsymbol{X}_2^l\boldsymbol{M}.\end{aligned} \tag{89}$$

Let us also define

$$\begin{aligned}&\boldsymbol{W}_1(l_1, l_2, l_3, l_4, k, j) \\ &\quad = \text{vec}(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1+\delta_k}\boldsymbol{X}^{l_2+\delta_k}\boldsymbol{M})\text{vec}^T(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_3+\delta_j}\boldsymbol{X}^{l_4+\delta_j}\boldsymbol{M}) \\ &\boldsymbol{W}_2(l_1, l_2, l_3, l_4, k, j) \\ &\quad = \text{vec}(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1+\delta_k}\boldsymbol{X}^{l_2+\delta_k}\boldsymbol{M}) \\ &\quad\quad \cdot \text{vec}^T(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_3}\boldsymbol{X}^{P-\delta_j}\tilde{\boldsymbol{X}}^{P-\delta_j}\boldsymbol{X}^{l_4}\boldsymbol{M}) \\ &\boldsymbol{W}_3(l_1, l_2, l_3, l_4, k, j) \\ &\quad = \text{vec}(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1}\boldsymbol{X}^{P-\delta_k}\tilde{\boldsymbol{X}}^{P-\delta_k}\boldsymbol{X}^{l_2}\boldsymbol{M}) \\ &\quad\quad \cdot \text{vec}^T(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_3+\delta_j}\boldsymbol{X}^{l_4+\delta_j}\boldsymbol{M}) \\ &\boldsymbol{W}_4(l_1, l_2, l_3, l_4, k, j) \\ &\quad = \text{vec}(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_1}\boldsymbol{X}^{P-\delta_k}\tilde{\boldsymbol{X}}^{P-\delta_k}\boldsymbol{X}^{l_2}\boldsymbol{M}) \\ &\quad\quad \cdot \text{vec}^T(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{X}}^{l_3}\boldsymbol{X}^{P-\delta_j}\tilde{\boldsymbol{X}}^{P-\delta_j}\boldsymbol{X}^{l_4}\boldsymbol{M}).\end{aligned} \tag{90}$$

Then, for $k \neq j$, we similarly obtain (cf. Appendix B)

$$\begin{aligned}\boldsymbol{\Lambda}_{k,j} = &\sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \cdot [\boldsymbol{W}_1(l_1, l_1, l_2, l_2, k, j) + \boldsymbol{W}_2(l_1, l_1, l_2, l_2, k, j)] \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \cdot [\boldsymbol{W}_3(l_1, l_1, l_2, l_2, k, j) + \boldsymbol{W}_4(l_1, l_1, l_2, l_2, k, j)]\end{aligned} \tag{91}$$

whereas when $k = j$, we have

$$\begin{aligned}\boldsymbol{\Lambda}_{j,j} = &(m_{4c} - 2\sigma_c^4) \sum_{l=0}^{P-1} \cdot [\boldsymbol{W}_1(l, l, l, l, j, j) + \boldsymbol{W}_4(l, l, l, l, j, j)] \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \cdot [\boldsymbol{W}_1(l_1, l_1, l_2, l_2, j, j) + \boldsymbol{W}_1(l_1, l_2, l_1, l_2, j, j)] \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \cdot [\boldsymbol{W}_2(l_1, l_1, l_2, l_2, j, j) + \boldsymbol{W}_3(l_1, l_1, l_2, l_2, j, j)] \\ &+ \sigma_c^4 \sum_{l_1, l_2=0}^{P-1} \cdot [\boldsymbol{W}_4(l_1, l_1, l_2, l_2, j, j) + \boldsymbol{W}_4(l_1, l_2, l_1, l_2, j, j)].\end{aligned} \tag{92}$$

Once $\boldsymbol{\Lambda}$ and $\boldsymbol{\Delta}$ are precomputed from system parameters, matrix $\boldsymbol{U}$ can also be obtained according to (86). $\square$

REFERENCES

[1] S. E. Bensley and B. Aazhang, "Subspace-based channel estimation for code division multiple access communication systems," *IEEE Trans. Commun.*, vol. 44, pp. 1009–1020, Aug. 1996.

[2] D. R. Brillinger, *Time Series: Data Analysis and Theory*, expanded ed. San Francisco, CA: McGraw-Hill/Holden Day, 1981.

[3] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, pp. 1327–1343, Aug. 1990.

[4] R. D. DeGroat, "Noniterative subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, pp. 571–577, Mar. 1992.

[5] B. Friedlander and B. Porat, "Asymptotically optimal estimation of MA and ARMA parameters of non-Gaussian processes from higher-order moments," *IEEE Trans. Automat. Contr.*, vol. 35, pp. 27–35, Jan. 1990.

[6] D. Gesbert and A. Paulraj, "Blind multi-user linear detection of CDMA signals in frequency selective channels," in *Proc. 31st Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 29–Nov. 1 1997.

[7] G. B. Giannakis and S. D. Halford, "Asymptotically optimal blind fractionally spaced channel estimation and performance analysis," *IEEE Trans. Signal Processing*, vol. 45, pp. 1815–1830, July 1996.

[8] G. B. Giannakis and M. K. Tsatsanis, "A unifying maximum-likelihood view of cumulant and polyspectral measures for non-Gaussian signal classification and estimation," *IEEE Trans. Inform. Theory*, vol. 38, pp. 386–406, Mar. 1992.

[9] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, A. Weaver, and C. E. Wheatley, "On the capacity of a cellular CDMA system," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 303–311, May 1991.

[10] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed. San Diego, CA: Academic, 1985.

[11] U. Madhow and M. L. Honig, "MMSE interference suppression for direct-sequence, spread-spectrum CDMA," *IEEE Trans. Commun.*, vol. 42, pp. 3178–3188, Dec. 1994.

[12] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[13] M. Honig, U. Madhow, and S. Verdu, "Blind adaptive multiuser detection," *IEEE Trans. Inform. Theory*, vol. 41, pp. 944–960, July 1995.

[14] H. Liu and G. Xu, "A subspace method for signature waveform estimation in synchronous CDMA systems," *IEEE Trans. Commun.*, vol. 44, pp. 1346–1354, Oct. 1996.

[15] R. Lupas and S. Verdu, "Linear multiuser detectors for synchronous code-division multiple-access channels," *IEEE Trans. Inform. Theory*, vol. 35, pp. 123–136, Jan. 1989.

[16] U. Mitra and H. V. Poor, "Analysis of an adaptive decorrelating detector for synchronous CDMA channels," *IEEE Trans. Commun.*, vol. 44, pp. 257–268, Feb. 1996.

[17] B. Porat, *Digital Processing of Random Signals: Theory and Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[18] J. Ramos and M. Zoltowski, "Reduced complexity blind 2D RAKE receiver for CDMA," in *Proc. 8th IEEE SP Workshop Stat. Signal Array Process.*, Corfu, Greece, June 1996, pp. 502–505.

[19] T. S. Rappaport, *Wireless Communications*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[20] G. W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, pp. 1535–1541, June 1992.

[21] E. G. Ström, S. Parkval, S. L. Miller, and B. E. Ottersten, "Propagation delay estimation in asynchronous direct-sequence code-division multiple access systems," *IEEE Trans. Commun.*, vol. 44, pp. 84–93, Jan. 1996.

[22] L. Tong, G. Xu, and T. Kailath, "Blind identification and equalization based on second-order statistics: A time-domain approach," *IEEE Trans. Inform. Theory*, vol. 40, pp. 340–349, Mar. 1994.

[23] M. Torlak and G. Xu, "Blind multiuser channel estimation in asynchronous CDMA systems," *IEEE Trans. Signal Processing*, vol. 45, pp. 137–147, Jan. 1997.

[24] M. Torlak, B. L. Evans, and G. Xu, "Blind estimation of FIR channels in CDMA systems with aperiodic spreading sequences," in *Proc. 31th Asilomar Conf. Signals, Syst., Comput.*, vol. 1, Pacific Grove, CA, Nov. 2–5, 1997, pp. 495–499.

[25] M. K. Tsatsanis, "Inverse filtering criteria for CDMA systems," *IEEE Trans. Signal Processing*, vol. 45, pp. 102–112, Jan. 1997.

[26] M. K. Tsatsanis and G. B. Giannakis, "Optimal decorrelating receivers for DS-CDMA systems: A signal processing framework," *IEEE Trans. Signal Processing*, vol. 44, pp. 3044–3055, Dec. 1996.

[27] ——, "Equalization of rapidly fading channels: Self-recovering methods," *IEEE Trans. Commun.*, vol. 44, pp. 619–630, May 1996.

[28] J. K. Tugnait, "Identification of linear stochastic systems via second order and fourth-order cumulant matching," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 393–407, May 1987.

[29] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[30] S. Verdu, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85–96, Jan. 1986.

[31] A. J. Weiss and B. Friedlander, "CDMA downlink channel estimation with aperiodic spreading," in *Proc. 31st Asilomar Conf. Signals, Syst., Comput.*, vol. 1, Pacific Grove, CA, Nov. 2–5, 1997, pp. 828–832.

Zhengyuan (Daniel) Xu received the B.E. and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1989 and 1991, respectively. He received the Ph.D. degree in electrical engineering from Stevens Institute of Technology, Hoboken, NJ, in May 1999.

From 1991 to 1996, he worked as a System Engineer and Department Manager at the Tsinghua Unisplendour Group Corporation, Tsinghua University. In August 1996, he joined Stevens Institute of Technology, where he was a Research Assistant working on signal processing for wireless communications, especially for multiuser CDMA systems. After receiving the Ph.D. degree, he worked as a Research Associate and then joined the Department of Electrical Engineering, University of California, Riverside, as an Assistant Professor. His current research interests include digital communication theory, multiuser detection, system parameter estimation, and high order statistics.

Dr. Xu received the Outstanding Student Award and Motorola Scholarship from Tsinghua University and the Peskin Award for his outstanding performance at Stevens Institute of Techology.

Michail K. Tsatsanis (M'93) received the diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1987, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Virginia, Charlottesville, in 1990 and 1993, respectively.

He is currently an Associate Professor with the Electrical and Computer Engineering Department, Stevens Institute of Technology, Hoboken, NJ. His general research interests lie in the areas of statistical signal and array processing, system identification, pattern recognition, higher order statistics, and wavelet theory. His current interests focus on signal processing techniques for wireless communications including blind equalization, multiuser detection, fading channel estimation and tracking, and signal processing methods for networking problems.

Dr. Tsatsanis is a Member of the IEEE Technical Committee on SP for Communications. He has served as a Member of the Organizing Committee for the 1996 IEEE Signal Processing Workshop on SSAP and is the Technical Co-chair of the Organizing Committee for the 1999 IEEE Workshop on Signal Processing Advances on Wireless Communications. He received the 1998 NSF CAREER award and the 1998 SP Society Young Author Best Paper Award. He is an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE COMMUNICATIONS LETTERS.