Decoupled Multiuser Code-Timing Estimation for Code-Division Multiple-Access Communication Systems

Hongbin Li, *Member, IEEE*, Jian Li, *Senior Member, IEEE*, and Scott L. Miller, *Senior Member, IEEE*

***Abstract*—We present herein a decoupled multiuser acquisition (DEMA) algorithm for code-timing estimation in asynchronous code-division multiple-access (CDMA) communication systems. The DEMA estimator is an asymptotic (for large data samples) maximum-likelihood method that models the channel parameters as deterministic unknowns. By evoking the mild assumption that the transmitted data bits for all users are independently and identically distributed, we show that the multiuser timing estimation problem that usually requires a search over a multidimensional parameter space decouples into a set of noniterative one-dimensional problems. Hence, the proposed algorithm is computationally efficient. DEMA has the desired property that, in the absence of noise, it obtains the *exact* parameter estimates even with finite number of data samples which can be heavily correlated. Another important feature of DEMA is that it exploits the structure of the receiver vectors and, therefore, is near–far resistant. Numerical examples are included to demonstrate and compare the performances of DEMA and a few other standard code-timing estimators.**

***Index Terms*—Code-division multiaccess, delay estimation, maximum-likelihood estimation, multiuser channel estimation.**

I. Introduction

DIRECT-SEQUENCE code-division multiple access (DS-CDMA) has been considered among the most promising multiplexing technologies for cellular telecommunication services. A drawback intrinsic to DS-CDMA systems is the so-called near–far problem, i.e., the signal from a distant desired user is likely to be overwhelmed by the strong signals from nearby interferers. Many near–far resistant receivers have been proposed (see, e.g., [1]–[4] and references therein), most of which assume the exact knowledge of one or several parameters such as the code timing, received power, and carrier phase for each user. These parameters are typically unknown and need to be estimated in a practical system. In this paper, we address the problem of code-timing estimation. Given accurate code-timing estimates, there is a wealth of good methods for estimating the other parameters (see, e.g., [5]–[8]).

A standard technique for code acquisition is the correlator [9]. It is well known that the correlator coincides with the optimal maximum-likelihood (ML) method for a single user in the presence of white Gaussian noise but its performance degrades drastically in a near–far multiuser environment. A modified correlator-like technique was developed based on the minimum mean-squared-error (MMSE) receiver [10]. The MMSE timing estimator attains better near–far resistance at the cost of some increased complexity. An alternative near–far resistant code-timing estimator is the MUSIC algorithm [11], [12], originally proposed by Schmidt [13] for direction-of-arrival estimation in array processing. Unlike the other techniques discussed herein, the MUSIC algorithm is *blind* in the sense that it needs no training for the purpose of code acquisition. However, the MUSIC timing estimator is computationally involved and its subscriber capacity is fairly restrictive. While most code-timing estimators are formulated in the time domain, Zhang *et al.* proposed a frequency domain based technique for code-timing estimation [14]. The key idea therein is to design training sequences that are orthogonal in the frequency domain so that different user signals, after being Fourier transformed, can be separated from one another. In [15] and [16], Chang and Chen investigated the issue of joint code-timing and carrier phase estimation for DS-CDMA systems.

Motivated by the work in [17], another interesting code-timing estimator, which is referred to as the large sample maximum-likelihood (LSML) algorithm, was recently introduced in [18]. LSML is found to be able to accommodate more users than most existing methods while maintaining a good acquisition performance and timing estimation accuracy. The derivation of the LSML algorithm relies on a receiver vector model whereby the code sequence and the transmitted data bits for only the desired user are assumed known whereas the signals from the interfering users and noise are modeled as unknown colored Gaussian noise. An advantage of treating the interfering signals in such an unstructured manner is that it facilitates the suppression of not only multiple-access interference (MAI), but cochannel narrow-band interferences as well.

In this paper, we present a decoupled multiuser acquisition (DEMA) algorithm for code-timing estimation. As a multiuser code-timing estimator, the DEMA algorithm requires the knowledge of the code sequences and the transmitted data bits

Paper approved by M. Brandt-Pearce, the Editor for Modulation and Signal Design of the IEEE Communications Society. Manuscript received December 17, 1998; revised October 21, 1999, April 27, 2000, and September 26, 2000. This work was supported in part by the New Jersey Commission on Science and Technology, the National Science Foundation under Grant and Grant NCR 9628642, and the Office of Naval Research under Grant N00014-96-0817. This paper was presented in part at the International Conference on Broadband Wireless Access Systems, San Francisco, CA, December 2000.

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

J. Li is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: li@dsp.ufl.edu).

S. L. Miller is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA.

Publisher Item Identifier S 0090-6778(01)06944-6.

for all users. Note that assuming the knowledge of the code sequences does not introduce any restriction in the reverse (mobile to base) link since the base station assigns the codes to the mobile users when they enter the cell. The knowledge of the transmitted data bits for all users may appear restrictive at a first sight. Yet, this can be achieved by using a training sequence for each new user whose timing is to be determined, similar to the MMSE or LSML timing estimator. For the remaining users which are in the stage of normal transmission and whose timing should be known to the base station, we may assume that their data bits can be reliably detected (through the use of error-correction coding) at the base station and the estimated data bits can be used by the DEMA algorithm as if they were the true transmitted data bits. (The case of imperfect knowledge of the data bits is also investigated in Section IV.) As a result of using such (perhaps readily available) information, DEMA is extremely near–far resistant and allows a system to be fully loaded with little performance degradation.

The DEMA algorithm is derived by taking all users into account so that the structure of the received signal can be fully exploited. With the assumption that the transmitted bits for all users are independently and identically distributed (i.i.d.), we show that DEMA is a decoupled algorithm which obtains the timing estimates for all users simultaneously and efficiently. We also show that the DEMA algorithm is asymptotically (for large data samples) equivalent to the optimal ML method. In the absence of noise, DEMA yields the *exact* parameter estimates with finite number of data bits which may or may not be correlated with one another.

The remainder of this paper is organized as follows. In Section II, we describe the data model and formulate the problem under investigation. In Section III, we derive the DEMA algorithm. Section IV contains the numerical examples. Finally, the study is summarized in Section V.

II. Data Model and Problem Formulation

The system under investigation is an asynchronous K-user DS-CDMA system using binary phase-shift keying (BPSK) modulation. The transmitted signal for the kth user has the form

$$x_k(t) = \sqrt{2P_k}s_k(t)\cos(\omega_c t + \theta'_k), \quad k = 1, 2, \ldots, K \quad (1)$$

where

- P_k kth user's transmitted power;
- ω_c carrier frequency;
- θ'_k random carrier phase uniformly distributed over the interval $[0, 2\pi)$.

$s_k(t)$ in (1) is the baseband signal of the kth user having the form

$$s_k(t) = \sum_{m=0}^{M-1} d_k(m)c_k(t - mT_b) \quad (2)$$

where

- M number of bits considered for code acquisition;
- T_b data bit interval;
- $d_k(m)$ mth transmitted data bit;
- $c_k(t)$ spreading waveform.

$$c_k(t) = \sum_{n=0}^{N-1} c'_k(n)\Pi(t - nT_c) \quad (3)$$

in which $c'_k(n) \in \{\pm 1\}$, $N = T_b/T_c$, and $\Pi(t)$ denotes a unit rectangular pulse over the chip period $[0, T_c)$.

For the case of flat-fading, the received signal can be written as

$$y(t) = \sum_{k=1}^{K} a'_k x_k(t - \tau_k) + n(t) \quad (4)$$

where a'_k and τ_k denote the fading coefficient and, respectively, the propagation delay for the kth user $n(t)$ denotes the channel noise, assumed to be zero-mean white Gaussian. Similar to other existing timing estimators that utilize training sequences [10], [18], [19], it is assumed that the receiver and the transmitter have aligned their clocks roughly to within a bit interval, i.e., $\tau_k \in [0, T_b)$, $k = 1, 2, \ldots, K$. This may be achieved, for example, by using a side signaling channel for call setup [20]. It should be noted that in a picocellular or a quasi-synchronous (QS) CDMA system where it is guaranteed that the propagation delay will be within one bit interval (see, e.g., [21] and references therein), using a side channel for initial synchronization is unnecessary.

The receiver front-end consists of an in-phase quadrature (IQ) mixer followed by an integrate-and-dump filter (IDF) (see, e.g., [11]) with integration time $T_i = T_c/Q$, where the integer $Q \geq 1$ is called the oversampling factor. The received complex sequence, $\{y(l)\}$, can be expressed as (with double frequency terms ignored)

$$y(l) = \sum_{k=1}^{K} a_k \sqrt{P_k} e^{j\theta_k} \frac{1}{T_i} \int_{(l-1)T_i}^{lT_i} s_k(t - \tau_k)dt + n(l), \quad l = 0, 1, \ldots, MNQ - 1 \quad (5)$$

where $\theta_k = \theta'_k - \omega_c \tau_k$, $n(l)$ denotes the zero-mean complex white Gaussian noise with variance σ_n^2, and a_k is the fading coefficient, which is modeled as zero-mean complex Gaussian assuming the stationary Rayleigh fading channel model [20].

For any algorithm that involves using the IDF, the choice of the oversampling factor Q should be made by a tradeoff between algorithmic performance and computational complexity. When $Q = 1$, i.e., the IDF output is sampled at the chip rate, an integration interval of $T_i = T_c$ in general contains components from two adjacent chips for each user since the the received signal is chip-asynchronous. Averaging over adjacent chips attenuates the high frequency components of the spreading waveforms, leading to a signal-to-noise ratio (SNR) loss. A simple calculation shows that the worst-case loss in SNR for a particular user is 3 dB when the timing misalignment for that user is $0.5T_c$, and that an average loss in SNR is $10\log_{10}(3/2) = 1.76$ dB, assuming that the delay is uniformly distributed between 0 and T_b (also see [12, p. 1010]). The loss in SNR can be remedied by using $Q > 1$ (see Section IV for a numerical example comparing the performance of choosing different Q). However, the size of the received data grows proportionally as Q increases, and so does the computational complexity of the algorithm (see, e.g., Section III-C).

Let the received vector during the mth bit interval $\mathbf{y}(m)$ be defined as

$$\begin{aligned}\mathbf{y}(m) &= [y(mNQ+1) \quad y(mNQ+2) \quad \dots \quad y(mNQ+NQ)]^{\mathrm{T}} \\ &\in \mathbb{C}^{NQ\times 1}\end{aligned} \tag{6}$$

where $(\cdot)^{\mathrm{T}}$ denotes the transpose, and the noise vector, $\mathbf{n}(m) \in \mathbb{C}^{NQ\times 1}$, be similarly formed from $\{n(l)\}$. Let

$$\mathbf{c}_k = [\, c_k(0) \quad c_k(1) \quad \dots \quad c_k(NQ-1)\,]^{\mathrm{T}} \in \{\pm 1\}^{NQ\times 1} \tag{7}$$

where $c_k(n) = (1/T_i)\int_{(n-1)T_i}^{nT_i} c_k(t)dt$. Then, the received vectors $\{\mathbf{y}(m)\}$ can be written as [11], [12], [18]

$$\mathbf{y}(m) = \sum_{k=1}^{K} \beta_k \mathbf{A}_k(\tau_k)\mathbf{z}_k(m) + \mathbf{n}(m), \qquad m = 0, 1, \dots, M-1 \tag{8}$$

where

$$\beta_k = a_k\sqrt{P_k}e^{j\theta_k} \tag{9}$$

$$\mathbf{A}_k(\tau_k) = \left[\, \mathbf{a}_k^{(1)}(\tau_k) \quad \mathbf{a}_k^{(2)}(\tau_k) \,\right] \in \mathbb{R}^{NQ\times 2} \tag{10}$$

and

$$\mathbf{z}_k(m) = [\, d_k(m-1) \quad d_k(m)\,]^{\mathrm{T}} \in \{\pm 1\}^{2\times 1}. \tag{11}$$

Note that $\mathbf{z}_k(0)$ is not defined since $d_k(-1)$ is unknown. We can arbitrarily choose $d_k(-1) = 0$ or discard the observation vector $\mathbf{y}(0)$, which will have little effect on the the estimator to be derived in the sequel if M is not too small. Let $\tau_k = (p_k + \mu_k)T_i$, where $p_k \in \{0,\ 1,\ \dots,\ NQ-1\}$ and $0 \le \mu_k < 1$. Then, $\mathbf{a}_k^{(1)}(\tau_k)$ and $\mathbf{a}_k^{(2)}(\tau_k)$ in (10) are given by

$$\mathbf{a}_k^{(1)}(\tau_k) = [(1-\mu_k)\mathbf{P}_1(p_k) + \mu_k\mathbf{P}_1(p_k+1)]\,\mathbf{c}_k \in \mathbb{R}^{NQ\times 1} \tag{12}$$

and

$$\mathbf{a}_k^{(2)}(\tau_k) = [(1-\mu_k)\mathbf{P}_2(p_k) + \mu_k\mathbf{P}_2(p_k+1)]\,\mathbf{c}_k \in \mathbb{R}^{NQ\times 1} \tag{13}$$

where $\mathbf{P}_1(p)$ and $\mathbf{P}_2(p)$ denote the $NQ \times NQ$ shifting matrices

$$\mathbf{P}_1(p) \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I}_p \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad \mathbf{P}_2(p) \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{NQ-p} & \mathbf{0} \end{bmatrix}. \tag{14}$$

Hereafter, $\mathbf{I}_p$ denotes the $p \times p$ identity matrix. To facilitate our derivation, we rewrite (8) more compactly as

$$\mathbf{y}(m) = \mathbf{B}\mathbf{s}(m) + \mathbf{n}(m) \tag{15}$$

where

$$\mathbf{B} = [\beta_1\mathbf{A}_1(\tau_1) \quad \beta_2\mathbf{A}_2(\tau_2) \quad \dots \quad \beta_K\mathbf{A}_K(\tau_K)] \in \mathbb{C}^{NQ\times 2K} \tag{16}$$

and

$$\mathbf{s}(m) = [\,\mathbf{z}_1^{\mathrm{T}}(m) \quad \mathbf{z}_2^{\mathrm{T}}(m) \quad \dots \quad \mathbf{z}_K^{\mathrm{T}}(m)\,]^{\mathrm{T}} \in \{\pm 1\}^{2K\times 1}. \tag{17}$$

The derivation of the DEMA algorithm in Section III makes use of a few additional assumptions. Specifically, we assume that the code sequences and the data bits for all users are known. Let

$$\mathbf{R}_{\mathbf{ss}}(M) = \frac{1}{M}\sum_{m=0}^{M-1} \mathbf{s}(m)\mathbf{s}^{\mathrm{T}}(m). \tag{18}$$

We assume the limiting matrix $\mathbf{R}_{\mathbf{ss}}$, defined by

$$\mathbf{R}_{\mathbf{ss}} = \lim_{M\to\infty} \mathbf{R}_{\mathbf{ss}}(M) \tag{19}$$

exists. We further assume that the data bits for all user are i.i.d. so that

$$\mathbf{R}_{\mathbf{ss}} = \mathbf{I}_{2K}. \tag{20}$$

Finally, it is assumed that $\mathbf{s}(m)$ and $\mathbf{n}(m)$ are uncorrelated, i.e.,

$$\lim_{M\to\infty} \frac{1}{M}\sum_{m=0}^{M-1} \mathbf{s}(m)\mathbf{n}^{\mathrm{H}}(m) = \mathbf{0}, \quad \text{with probability 1 (w.p. 1.)} \tag{21}$$

where $(\cdot)^{\mathrm{H}}$ denotes the Hermitian transpose.

The problem of interest is to estimate $\{\tau_k\}_{k=1}^{K}$, or equivalently, $\{p_k, \mu_k\}_{k=1}^{K}$, from the measurements $\{\mathbf{y}(m)\}_{m=0}^{M-1}$.

III. Code-Timing Estimation Algorithms

We relate our derivation of the DEMA algorithm to that of the LSML method to shed more light on the properties of the two timing estimators. To that end, we briefly discuss the LSML estimator before introducing the DEMA algorithm. A computational complexity analysis of the two algorithms is also included in this section.

A. LSML

Even though LSML was not derived specifically for multiuser code-timing estimation, it can be used to solve this problem by estimating one user at a time. Without loss of generality, we assume that the first user is the desired user. Then, we can rewrite (8) as

$$\begin{aligned}\mathbf{y}(m) &= \beta_1\mathbf{A}_1(\tau_1)\mathbf{z}_1(m) + \sum_{k=2}^{K}\beta_k\mathbf{A}_k(\tau_k)\mathbf{z}_k(m) + \mathbf{n}(m) \\ &\triangleq \beta_1\mathbf{A}_1(\tau_1)\mathbf{z}_1(m) + \mathbf{e}(m).\end{aligned} \tag{22}$$

In the above, the observation noise and MAI are lumped together into an *unstructured* term $\mathbf{e}(m)$. The LSML algorithm is derived by modeling $\mathbf{e}(m)$ as the circularly symmetric complex Gaussian noise with zero-mean and unknown covariance matrix $\mathbf{Q}$ that satisfies

$$E\left[\mathbf{e}(m_i)\mathbf{e}^{\mathrm{H}}(m_j)\right] = \mathbf{Q}\delta_{i,j} \tag{23}$$

where $E[\cdot]$ denotes the expectation operator and $\delta_{i,j}$ denotes the Kronecker delta. The covariance matrix $\mathbf{Q}$ is estimated in an unstructured manner in [18] so that a whitening process is evoked to suppress MAI. Obviously, this unstructured approach also facilitates the suppression of cochannel narrow-band interferences, which can be included in $\mathbf{e}(m)$ and be suppressed. As such, the LSML algorithm allows some additional flexibility in its applications over other existing techniques. Note that (23) implies that $\mathbf{e}(m_i)$ and $\mathbf{e}(m_j)$ are uncorrelated for $i \neq j$. However, due to the asynchronous nature, $\mathbf{e}(m)$ and $\mathbf{e}(m+1)$ are usually correlated with each other. LSML ignores this correlation for the sake of yielding a simple estimator. The penalty is that the estimation accuracy in general degrades. (See [18] for more details of the LSML method.)

B. DEMA

DEMA is a better approach to the problem of interest, as described next. By treating $\{a_k, P_k, \theta_k, \tau_k\}_{k=1}^K$ as deterministic unknowns and observing that $\mathbf{n}(m)$ is circularly symmetric complex Gaussian with zero-mean and $E\{\mathbf{n}(m_i)\mathbf{n}^{\mathrm{H}}(m_j)\} = \sigma_n^2 \mathbf{I}_{NQ}\delta_{i,j}$, it is readily shown that the ML solution is equivalent to

$$\{\hat{\beta}_k, \hat{\tau}_k\}_{k=1}^K = \arg \min_{\{\beta_k, \tau_k\}_{k=1}^K} \mathrm{tr}\left\{ \frac{1}{M} \sum_{m=0}^{M-1} [\mathbf{y}(m) - \mathbf{B}\mathbf{s}(m)][\mathbf{y}(m) - \mathbf{B}\mathbf{s}(m)]^{\mathrm{H}} \right\} \quad (24)$$

where $\mathrm{tr}\{\cdot\}$ denotes the trace operator. Let

$$\mathbf{R}_{\mathbf{sy}}(M) = \frac{1}{M}\sum_{m=0}^{M-1} \mathbf{s}(m)\mathbf{y}^{\mathrm{H}}(m) \quad (25)$$

and $\mathbf{R}_{\mathbf{yy}}(M)$ be similarly defined from $\{\mathbf{y}(m)\}_{m=0}^{M-1}$ as $\mathbf{R}_{\mathbf{ss}}(M)$ is from $\{\mathbf{s}(m)\}_{m=0}^{M-1}$. Minimizing (24) with respect to $\mathbf{B}$ gives

$$\hat{\mathbf{B}} = \mathbf{R}_{\mathbf{sy}}^{\mathrm{H}}(M)\mathbf{R}_{\mathbf{ss}}^{-1}(M) \quad (26)$$

where we have assumed that $\mathbf{R}_{\mathbf{ss}}^{-1}(M)$ exists. Next, we rearrange the cost function in (24) as follows:

$$\begin{aligned} &\mathrm{tr}\left\{ \frac{1}{M} \sum_{m=0}^{M-1} [\mathbf{y}(m) - \mathbf{B}\mathbf{s}(m)][\mathbf{y}(m) - \mathbf{B}\mathbf{s}(m)]^{\mathrm{H}} \right\} \\ &= \mathrm{tr}\left\{ \mathbf{R}_{\mathbf{yy}}(M) - \mathbf{B}\mathbf{R}_{\mathbf{sy}}(M) - \mathbf{R}_{\mathbf{sy}}^{\mathrm{H}}(M)\mathbf{B}^{\mathrm{H}} + \mathbf{B}\mathbf{R}_{\mathbf{ss}}(M)\mathbf{B}^{\mathrm{H}} \right\} \\ &= \mathrm{tr}\left\{ \mathbf{R}_{\mathbf{yy}}(M) - \hat{\mathbf{B}}\mathbf{R}_{\mathbf{ss}}(M)\hat{\mathbf{B}}^{\mathrm{H}} \right\} + \mathrm{tr}\left\{ (\mathbf{B} - \hat{\mathbf{B}})\mathbf{R}_{\mathbf{ss}}(M)(\mathbf{B} - \hat{\mathbf{B}})^{\mathrm{H}} \right\}. \end{aligned} \quad (27)$$

Since the first term of (27) is independent of $\mathbf{B}$, minimizing (27) reduces to

$$\{\hat{\beta}_k, \hat{\tau}_k\}_{k=1}^K = \arg \min_{\{\beta_k, \tau_k\}_{k=1}^K} \mathrm{tr}\left\{ (\mathbf{B} - \hat{\mathbf{B}})\mathbf{R}_{\mathbf{ss}}(M)(\mathbf{B} - \hat{\mathbf{B}})^{\mathrm{H}} \right\}. \quad (28)$$

The exact ML estimates of the unknown parameters are obtained by minimizing the cost function of (28), which in general requires a search over a $3K$-dimensional parameter space (note that β_k is complex-valued) and is computationally prohibitive. In the following we derive the DEMA algorithm which coincides asymptotically (for large M) with the exact ML method, but at a significantly reduced computational complexity.

The key idea of the DEMA algorithm involves exploiting the structure of $\mathbf{R}_{\mathbf{ss}} = \mathbf{I}_{2K}$. Observe that $\mathbf{R}_{\mathbf{ss}}(M)$ is a $(1/\sqrt{M})$-consistent estimate of $\mathbf{R}_{\mathbf{ss}}$[22]. Likewise, by using the assumption that the data bits and the noise samples are independent of each other, it is readily shown that $\hat{\mathbf{B}}$ is also a $(1/\sqrt{M})$-consistent estimate of $\mathbf{B}$. It follows that

$$\begin{aligned} &\mathrm{tr}\left\{ (\mathbf{B} - \hat{\mathbf{B}})\mathbf{R}_{\mathbf{ss}}(M)(\mathbf{B} - \hat{\mathbf{B}})^{\mathrm{H}} \right\} \\ &\quad = \mathrm{tr}\left\{ (\mathbf{B} - \hat{\mathbf{B}})\mathbf{R}_{\mathbf{ss}}(\mathbf{B} - \hat{\mathbf{B}})^{\mathrm{H}} \right\} + O\left(M^{-3/2}\right). \end{aligned} \quad (29)$$

The above equality indicates that to within a second-order approximation, $\mathbf{R}_{\mathbf{ss}}(M)$ in (28) can be replaced by $\mathbf{R}_{\mathbf{ss}}$ without affecting the asymptotic performance of the parameter estimates. Hence, the solution given in (28) is asymptotically (for large M) equivalent to

$$\{\hat{\beta}_k, \hat{\tau}_k\}_{k=1}^K = \arg \min_{\{\beta_k, \tau_k\}_{k=1}^K} \mathrm{tr}\left\{ \mathbf{R}_{\mathbf{ss}}(\mathbf{B} - \hat{\mathbf{B}})^{\mathrm{H}}(\mathbf{B} - \hat{\mathbf{B}}) \right\} \quad (30)$$

where we have made use of the identity $\mathrm{tr}\{\mathbf{AB}\} = \mathrm{tr}\{\mathbf{BA}\}$ for any matrices $\mathbf{A}$ and $\mathbf{B}$ of compatible dimensions. Let

$$\hat{\mathbf{B}} = [\,\hat{\mathbf{B}}_1 \quad \hat{\mathbf{B}}_2 \quad \ldots \quad \hat{\mathbf{B}}_K\,]. \quad (31)$$

The fact that $\mathbf{R}_{\mathbf{ss}} = \mathbf{I}_{2K}$ decouples (30) into a series of K minimization problems

$$\{\hat{\beta}_k, \hat{\tau}_k\} = \arg \min_{\beta_k, \tau_k} \left\| \beta_k \mathbf{A}_k(\tau_k) - \hat{\mathbf{B}}_k \right\|_F^2, \quad k = 1, 2, \ldots, K \quad (32)$$

where $\|\cdot\|_F$ denotes the Frobenius norm [23]. Let

$$\mathbf{a}_k(\tau_k) = \mathrm{vec}[\mathbf{A}_k(\tau_k)] \in \mathbb{R}^{2NQ\times 1} \quad (33)$$

and

$$\hat{\mathbf{b}}_k = \mathrm{vec}[\hat{\mathbf{B}}_k] \in \mathbb{C}^{2NQ\times 1} \quad (34)$$

where $\mathrm{vec}[\cdot]$ denotes the operation of stacking the columns of a matrix on top of one another. Then, minimizing the cost function in (32) with respect to τ_k and β_k yields

$$\hat{\tau}_k = \arg\max_{\tau_k} \frac{\left|\mathbf{a}_k^{\mathrm{T}}(\tau_k)\hat{\mathbf{b}}_k\right|^2}{\mathbf{a}_k^{\mathrm{T}}(\tau_k)\mathbf{a}_k(\tau_k)} \quad (35)$$

and

$$\hat{\beta}_k = \frac{\mathbf{a}_k^{\mathrm{T}}(\hat{\tau}_k)\hat{\mathbf{b}}_k}{\mathbf{a}_k^{\mathrm{T}}(\hat{\tau}_k)\mathbf{a}_k(\hat{\tau}_k)}. \quad (36)$$

As a by-product, the amplitude and the carrier phase can be estimated as $|\hat{\beta}_k|$ and $\arg(\hat{\beta}_k)$, respectively, once $\hat{\tau}_k$ and $\hat{\beta}_k$ are obtained [see (9)].

The maximization of (35) is simple to perform. We first rewrite (10) as

$$\mathbf{A}_k(\tau_k) = \left[\, \mathbf{a}_k^{(1)}(\tau_k) \quad \mathbf{a}_k^{(2)}(\tau_k) \,\right] = \left[\, \mathbf{F}_k^{(1)}\boldsymbol{\mu}_k \quad \mathbf{F}_k^{(2)}\boldsymbol{\mu}_k \,\right] \quad (37)$$

where

$$\mathbf{F}_k^{(1)} = [\,\mathbf{P}_1(p_k)\mathbf{c}_k \quad \mathbf{P}_1(p_k+1)\mathbf{c}_k\,] \in \{0, \pm 1\}^{NQ\times 2} \quad (38)$$

$$\mathbf{F}_k^{(2)} = [\,\mathbf{P}_2(p_k)\mathbf{c}_k \quad \mathbf{P}_2(p_k+1)\mathbf{c}_k\,] \in \{0, \pm 1\}^{NQ\times 2} \quad (39)$$

TABLE I
SUMMARY OF THE DEMA ALGORITHM

Step 1:
1) $\mathbf{R}_{\mathbf{sy}}(M) = \frac{1}{M}\sum_{m=0}^{M-1}\mathbf{s}(m)\mathbf{y}^H(m)$. $\Rightarrow O(KMNQ)$ flops
2) $\mathbf{R}_{\mathbf{ss}}(M) = \frac{1}{M}\sum_{m=0}^{M-1}\mathbf{s}(m)\mathbf{s}^T(m)$ $\Rightarrow O(K^2M)$ flops
3) Compute $\hat{\mathbf{B}} = \mathbf{R}_{\mathbf{sy}}^H(M)\mathbf{R}_{\mathbf{ss}}^{-1}(M)$ by solving $\mathbf{R}_{\mathbf{sh}}^H(M) = \hat{\mathbf{B}}\mathbf{R}_{\mathbf{ss}}(M)$. $\Rightarrow O(K^3) + O(K^2NQ)$ flops

Step 2: For $k = 1 : K$
- 1) Form $\hat{\mathbf{B}}_k$ according to (31) and $\hat{\mathbf{b}}_k = \vec{[\hat{\mathbf{B}}_k]}$. $\Rightarrow 0$ flops
- 2) Let $\mathcal{S}_k$ be a null set.
  For $p_k = 0 : NQ-1$
  - a) Obtain $\mathbf{F}_k$ by (41). $\Rightarrow 0$ flops
  - b) $\mathbf{U}_k = \Re(\mathbf{F}_k^T\hat{\mathbf{b}}_k\hat{\mathbf{b}}_k^H\mathbf{F}_k)$ and $\mathbf{V}_k = \mathbf{F}_k^T\mathbf{F}_k$. $\Rightarrow O(NQ)$ flops
  - c) Compute the coefficients of the polynomials $N(p_k,\mu_k)$, $D(p_k,\mu_k)$, and $R(p_k,\mu_k)$ by (42) and (45). $\Rightarrow O(1)$ flops
  - d) Include $R(p_k, 0)$ in $\mathcal{S}_k$.
  - e) Find the zeros, μ'_{k_1} and μ'_{k_2}, of $S(p_k,\mu_k)$, according to (45). $\Rightarrow O(1)$ flops
  - f) If $0 < \mu'_{k_i} < 1, i = 1$ or 2, then include $R(p_k, \mu'_{k_i})$ in $\mathcal{S}_k$.

  end
- 3) Choose $\hat{\tau}_k = (\hat{p}_k + \hat{\mu}_k)T_c$ which corresponds to the largest element of $\mathcal{S}_k$.

end

and

$$\boldsymbol{\mu}_k = [1-\mu_k \quad \mu_k]^{\mathrm{T}} \in \mathbb{R}^{2\times 1}. \tag{40}$$

It follows that

$$\mathbf{a}_k(\tau_k) = \begin{bmatrix}\mathbf{F}_k^{(1)} \\ \mathbf{F}_k^{(2)}\end{bmatrix}\boldsymbol{\mu}_k \triangleq \mathbf{F}_k\boldsymbol{\mu}_k. \tag{41}$$

Insertion of (41) into the cost function in (35) yields

$$R(p_k,\mu_k) \triangleq \frac{N(p_k,\mu_k)}{D(p_k,\mu_k)} \triangleq \frac{\boldsymbol{\mu}_k^{\mathrm{T}}(\mathbf{F}_k^{\mathrm{T}}\hat{\mathbf{b}}_k\mathbf{b}_k^{\mathrm{H}}\mathbf{F}_k)\boldsymbol{\mu}_k}{\boldsymbol{\mu}_k^{\mathrm{T}}\left(\mathbf{F}_k^{\mathrm{T}}\mathbf{F}_k\right)\boldsymbol{\mu}_k} = \frac{\boldsymbol{\mu}_k^{\mathrm{T}}\mathbf{U}_k\boldsymbol{\mu}_k}{\boldsymbol{\mu}_k^{\mathrm{T}}\mathbf{V}_k\boldsymbol{\mu}_k} \tag{42}$$

where

$$\mathbf{U}_k = \Re\left(\mathbf{F}_k^{\mathrm{T}}\hat{\mathbf{b}}_k\hat{\mathbf{b}}_k^{\mathrm{H}}\mathbf{F}_k\right) \in \mathbb{R}^{2\times 2} \tag{43}$$

and

$$\mathbf{V}_k = \mathbf{F}_k^{\mathrm{T}}\mathbf{F}_k \in \mathbb{Z}^{2\times 2}. \tag{44}$$

In the last equality of (42), we have used the fact that $\mathbf{x}^{\mathrm{T}}\Im(\mathbf{H})\mathbf{x} \equiv 0$ for any real-valued vector $\mathbf{x}$ and any Hermitian matrix $\mathbf{H}$ of proper dimensions. The maximization of $R(p_k,\mu_k)$ is similar to that in [18], as briefed next. Given $p_k \in \{0,1,\ldots,NQ-1\}$, $R(p_k,\mu'_k)$ may be a local maximum over the interval $(p_kT_i,(p_k+1)T_i)$ *only* if μ'_k is a stationary point, i.e., μ'_k is a zero of the polynomial

$$S(p_k,\mu_k) \triangleq D(p_k,\mu_k)\frac{\partial N(p_k,\mu_k)}{\partial \mu_k} - N(p_k,\mu_k)\frac{\partial D(p_k,\mu_k)}{\partial \mu_k}. \tag{45}$$

Both $N(p_k,\mu_k)$ and $D(p_k,\mu_k)$ are second-order polynomials of μ_k, and so is $S(p_k,\mu_k)$ (observe that the third-order terms cancel out). Consequently, for each p_k, μ'_k can be conveniently found by rooting a second-order polynomial. Note that $R(p_k,\mu_k)$ is generally not differentiable at the boundary points $\tau_k = p_kT_i$, $p_k = 0,1,\ldots,NQ-1$. Hence, $S(p_k,\mu_k)$ evaluated at these point may be local maxima as well. It is therefore guaranteed that the global maximum of $S(p_k,\mu_k)$ is attained at one of the stationary or boundary points and can be located by evaluating and comparing $S(p_k,\mu_k)$ at these points one by one. A summary of the DEMA algorithm is given in Table I.

Remark 1: It can be shown that the DEMA algorithm is an asymptotic (for large M) ML estimator and asymptotically achieves the Cramér–Rao bound (CRB) [17], the best performance bound of any unbiased estimator. Furthermore, DEMA is SNR consistent in the sense that, in the absence of noise, the DEMA estimates of the parameters are *exact* as long as $\mathbf{R}_{\mathbf{ss}}(M)$ has full rank. In that event, DEMA obtains the true values of the parameters even with a finite number of data bits $(M \geq 2K)$ which can be heavily correlated with one another. This is seen by observing that $\hat{\mathbf{B}}$ in (26) approaches the true $\mathbf{B}$ matrix defined in (16) when $\sigma_n^2 \to 0$.

Remark 2: The sufficient condition to apply the DEMA algorithm is the existence of $\mathbf{R}_{\mathbf{ss}}^{-1}(M)$. A necessary condition for the existence of $\mathbf{R}_{\mathbf{ss}}^{-1}(M)$ is $M \geq 2K$. Note that $\hat{\mathbf{R}}_{\mathbf{ss}}$ is a sum of M rank-1 matrices and, therefore, rank$(\mathbf{R}_{\mathbf{ss}}(M)) \leq M$. If $M < 2K$, then DEMA will not be able to yield valid parameter estimates. One solution is to use the Moore-Penrose pseudo-inverse [23] to compute $\mathbf{R}_{\mathbf{ss}}^{-1}(M)$. An alternative one is to replace $\mathbf{R}_{\mathbf{ss}}(M)$ with $\mathbf{I}_{2K}$, when $M < 2K$, since $\mathbf{R}_{\mathbf{ss}}(M)$ is a consistent estimate of $\mathbf{R}_{\mathbf{ss}} = \mathbf{I}_{2K}$. It should be stressed that either way the $\hat{\mathbf{B}}$ in (26) is no longer an ML estimate. Additionally, replacing $\mathbf{R}_{\mathbf{ss}}(M)$ with $\mathbf{I}_{2K}$ implies that the knowledge of the interfering users' data bits is not used, and the so-obtained timing estimator is reduced to some correlator-type method. Specifically, one can show that when the training bits for the kth user are 1's, replacing $\mathbf{R}_{\mathbf{ss}}(M)$ with $\mathbf{I}_{2K}$ in the DEMA algorithm leads to

$$\hat{\tau}_k = \arg\max_{\tau_k} \frac{\left| \left[\mathbf{a}_k^{(1)}(\tau_k) \right]^{\mathrm{T}} \left[\frac{1}{M} \sum_{m=0}^{M-1} \mathbf{y}(m) \right] \right|^2}{\mathbf{a}_k^{\mathrm{T}}(\tau_k)\mathbf{a}_k(\tau_k)} \qquad (46)$$

which is similar to the sliding correlator in [11].

Remark 3: For the sake of presentational simplicity, we have assumed rectangular chip waveforms in the above derivation. It is known that rectangular chip waveforms have infinite bandwidth and hence are not feasible in practical systems. However, the DEMA algorithm can be extended in a straightforward manner along the lines of [24] to the case of band-limited waveforms. The resulting DEMA algorithm will still be decoupled with respect to different users and, therefore, be computationally more attractive than the exact ML algorithm. The optimization of the modified DEMA cost function, however, will no longer consist of second-order polynomial rootings. Some nonlinear optimization routine will have to be used in general.

C. Computational Complexity

In the following, we briefly discuss the computational complexities of the DEMA and LSML algorithms. Complexity analysis of other methods, such as the MMSE and MUSIC timing estimators, can be found, for example, in [18].

We list the numbers of flops involved in each step of the DEMA algorithm in Table I. It is seen from Table I that for the case where only one user's timing is desired, DEMA requires $O(K^3 + K^2M + K^2NQ + KMNQ)$ flops. In a similar manner, we can show that the number of flops required by the LSML algorithm in the single-user estimation case is $O(N^3Q^3 + MN^2Q^2)$. If we assume that $N \approx M \approx K$ and $N \gg Q$, then DEMA and LSML will have similar computational complexities. Yet in most cases of interest, LSML is usually more involved than DEMA because of their different constant factors. For example, in a scenario where $N = 31$, $M = 50$, $K = 10$, and $Q = 1$, simulation results indicate that the number of flops required by LSML is about 4.5 times that by DEMA.

However, the difference becomes more striking for the multiuser estimation case. A K-user timing estimation by DEMA requires only $O(KN^2Q^2)$ flops in addition to those required for the single-user estimation case. (Hence, estimating the timing for one or K users by DEMA essentially has the same order of computational complexity.) On the other hand, LSML in the current case requires $O(KN^3Q^3)$ flops and, therefore, is an order of magnitude more involved than DEMA. To see this, recall that LSML estimates one user at a time. It treats all users other than the desired one as "colored noise" whose covariance matrix, the $\mathbf{Q}$ in (23), needs to be estimated. As such, $\mathbf{Q}$ is estimated and its inverse is computed K times, which leads to a computational load of $O(KN^3Q^3)$ flops. (For numerical stability, it has been a common practice to calculate a matrix inverse implicitly by solving some linear equation instead of calculating it explicitly. The two approaches, however, have similar computational complexity.) Additionally, it takes $O(N^2Q^2)$ flops to determine the LSML cost function (see (31) in [18]) and maximize it, as opposed to the $O(NQ)$ flops in obtaining and maximizing a similar cost function, the $R(p_k, \mu_k)$ in (42), in DEMA. In the multiuser estimation case, this LSML cost function is determined and maximized KNQ times, which gives rise to another computational load of $O(KN^3Q^3)$ flops.

IV. Numerical Results

In this section, we compare DEMA with two other methods that all require a training process, namely the LSML and MMSE-RLS timing estimators. MMSE-RLS stands for the MMSE timing estimator driven by the recursive least squares (RLS) algorithm, which was found to significantly outperform the MMSE estimator driven by the least mean squares (LMS) algorithm [18]. The correlator is not considered herein because of its well-known poor performance in a near–far multiuser environment; nor is the MUSIC timing estimator. The reader is referred to [18] for a numerical study of the correlator, LSML, MMSE-RLS, and MUSIC algorithms.

Each user is assigned a Gold sequence of $N = 31$. In the following, the timing estimate for one particular user is evaluated and compared, whose transmitted power, P_1, is scaled so that $P_1 = 1$. The other users are given a random received power with a log-normal distribution. The power of each interfering signal has a mean d dB (to be specified) above the desired user and a standard deviation of 10 dB, i.e., $P_k/P_1 = 10^{\xi_k/10}$, $k = 2, \ldots, K$, where $\xi_k \sim N(d, 100)$. The additive noise $n(t)$ is white Gaussian with zero-mean and power spectral density of $N_0/2$. The variance of the noise samples (normalized as indicated above), $n(l)$ in (5), is thus $\sigma_n^2 = NQN_0/E_{b1}$, where E_{b1} is the energy per bit for the first user. The timing offsets $\{\tau_k\}_{k=1}^K$ and the carrier phases $\{\theta_k\}_{k=1}^K$ are uniformly distributed over $[0, T_b)$ and, respectively, $[0, 2\pi)$. The fading coefficients $\{a_k\}_{k=1}^K$ are modeled as circularly symmetric complex Gaussian random variables with zero-mean and unit variance. The transmitted data bits for all users are equally likely to be $+1$ or -1 with the exception that the first user's training bits for MMSE-RLS are all 1's. (Note that we choose the MMSE timing estimator with all 1's training sequence because of its simplicity. The MMSE timing estimator can be modified to work with arbitrary training sequence. See, e.g., [25].) In what follows, the primary performance measure used is the probability of correct acquisition, which is defined to be the event $|\hat{\tau}_k - \tau_k| < T_c/2$. Another performance measure used is the root mean squared error (RMSE) of the timing estimate given correct acquisition. The results below are based on 500 Monte Carlo trials. The data bits and the parameters $\{\tau_k, \theta_k, P_k, a_k\}$ for all users (with the exception that $P_1 \equiv 1$ and the previous exception for MMSE-RLS just mentioned) are changed from one Monte Carlo run to another.

Performance Versus M: Let $K = 10$, $d = 10$, $E_{b1}/n_0 = 10$ dB, the oversampling factor $Q = 1$, and the number of data bits M be varied from 5 to 100. All the other parameters are as described above. Note that no timing estimates can be formed by LSML, when $M < N = 31$, nor by DEMA, when $M < 2K = 20$. For the latter case, we replace the singular $\hat{\mathbf{R}}_{\mathbf{ss}}$ by $(1/2)\mathbf{I}_{2K}$. Fig. 1 shows the probability of correct ac-

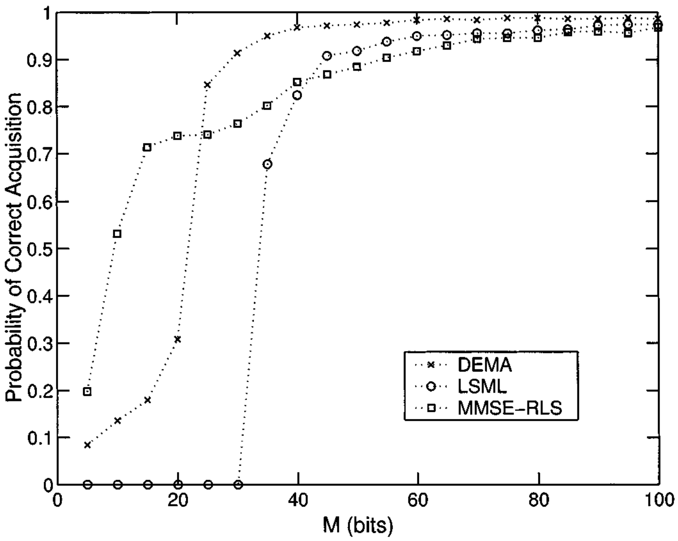

Fig. 1. Probability of correct acquisition versus M when $K = 10$, $N = 31$, $Q = 1$, $E_b/N_0 = 10$ dB, and $\xi_k \sim N(10, 100)$.

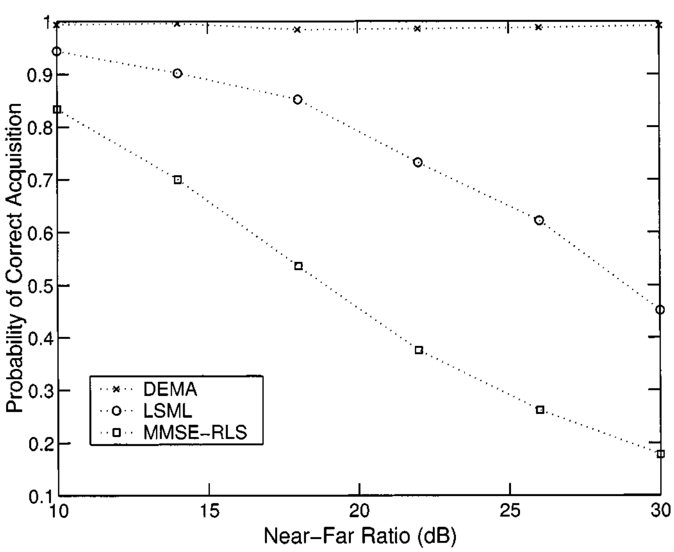

Fig. 2. Probability of correct acquisition versus near–far ratio when $K = 20$, $N = 31$, $M = 100$, $Q = 1$, and $E_b/N_0 = 10$ dB.

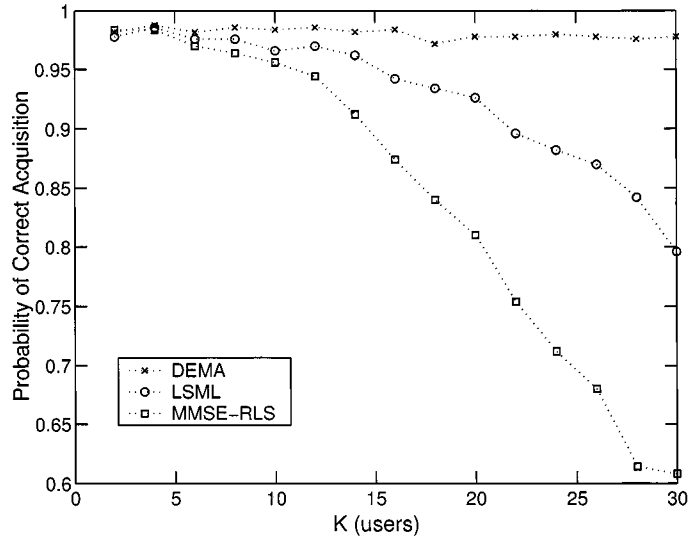

Fig. 3. Probability of correct acquisition versus K when $N = 31$, $M = 100$, $Q = 1$, $E_b/N_0 = 10$ dB, and $\xi_k \sim N(10, 100)$.

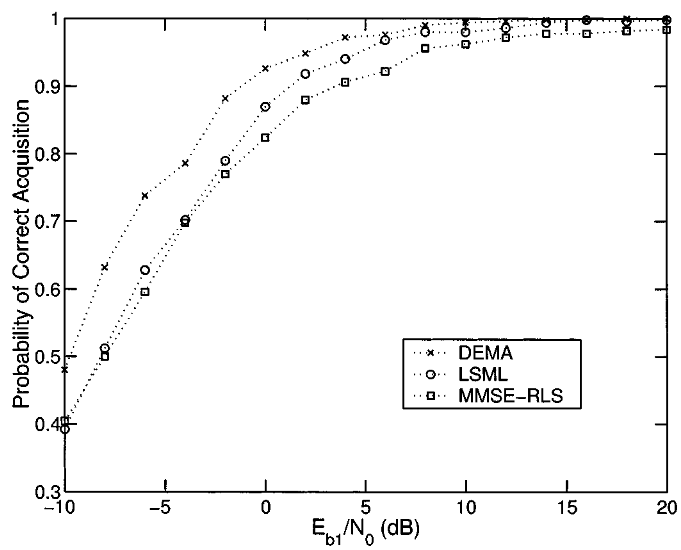

Fig. 4. Probability of correct acquisition versus E_{b_1}/N_0 when $K = 10$, $N = 31$, $Q = 1$, $M = 100$, and $\xi_k \sim N(10, 100)$.

quisition as M changes. As one can see, DEMA gives the best acquisition performance when $M \geq 25$. For small M, all estimators degrade significantly, with MMSE-RLS being slightly better than the others.

Performance Versus Near–Far Ratio: The near–far ratio is defined as the ratio of the mean of the random powers of the interfering users to that of the desired user. The parameters are the same as in the previous example except that $M = 100$, $K = 20$, and the near–far ratio (defined as d in decibels) is varied from 10 to 30 dB. As seen in Fig. 2, the near–far problem appears to have little effect on DEMA, whereas the performances of the other two, especially MMSE-RLS, degrade rapidly as the near–far ratio increases.

Performance Versus K: In this example we investigate the performance of the estimators as the number of users K varies. The parameters are similar to those in the first example with the exception that $M = 100$ and K is varied from 2 to 30. The results are shown in Fig. 3. It is seen that the performance of DEMA remains relatively unchanged, as opposed to the significant degradation of LSML and MMSE-RLS, as K increases.

Performance Versus E_{b_1}/N_0: Fig. 4 shows the probability of correct acquisition as E_{b_1}/N_0 varies from -10 to 20 dB. The other parameters are the same as those in the first example. We see that DEMA gives the best acquisition performance for all values of E_{b_1}/N_0 considered.

RMSE, CRB, and Bias: We now compare the RMSE of the timing estimates with the CRB. The CRB for the timing estimation problem is derived in the Appendix. Since the CRB is a function of $\{\tau_k, \theta_k, P_k, a_k\}$ and the data bits for all users, all these quantities are fixed in the Monte Carlo simulation in this example (i.e., only the additive noise is varied). Fig. 5(a) shows the RMSE of DEMA and LSML, and the CRB as M varies. We consider two cases, corresponding to the oversampling factor

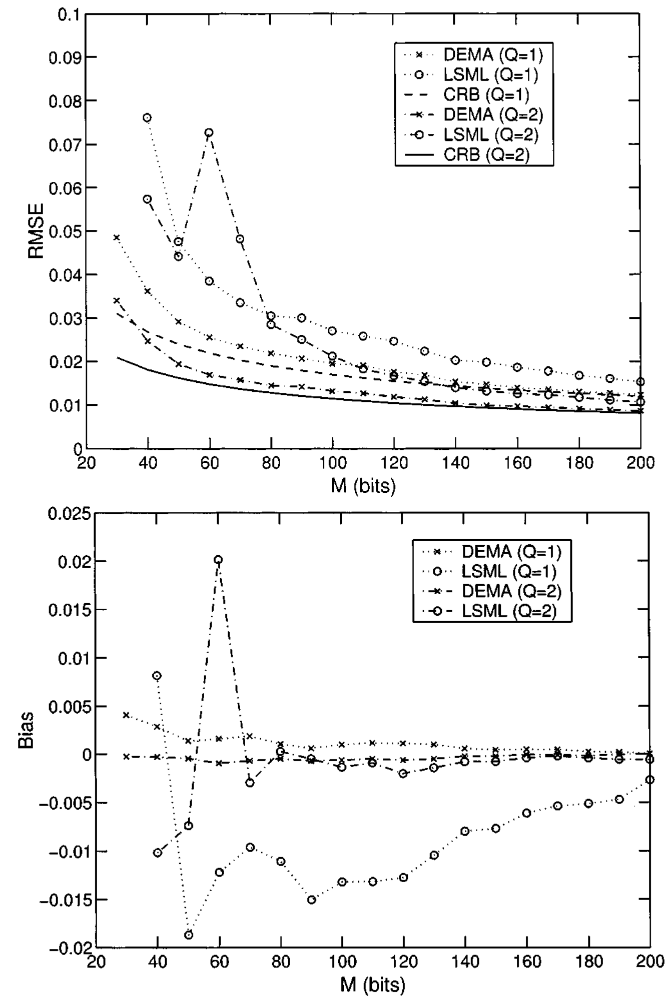

Fig. 5. RMSE and bias of the timing estimates, given correct acquisition, and the CRB versus M when $K = 10, N = 31, M = 100, E_b/N_0 = 10$ dB, and $\xi_k \sim N(10, 100)$. (The bias, RMSE and CRB are normalized with respect to T_c.) (a) RMSE and CRB. (b) Bias.

$Q = 1$ and $Q = 2$, respectively. Note that the CRB is a performance bound for unbiased estimators. It would therefore be of interest to know if DEMA is unbiased. The empirical bias of DEMA and LSML is shown in Fig. 5(b). Observe that increasing Q in general improves the performance of both DEMA and LSML in terms of RMSE and bias (except for the fluctuation experienced by LSML for relatively small M). Also observe that the CRB corresponding to $Q = 2$ is smaller than that corresponding to $Q = 1$. It is seen that the bias of DEMA is more than an order of magnitude smaller than its RMSE and, hence, DEMA appears to be unbiased even for finite M. It is also seen that for moderately small M (such as when $M = 50$), the RMSE of the DEMA timing estimates are very close to the CRB for both $Q = 1$ and $Q = 2$.

Data Bits Known Imperfectly: In the derivation of the DEMA algorithm, we have assumed that the data bits for all users are known perfectly. In this example we consider the case when some of the data bits from users in normal transmission are estimated incorrectly. We change the percentage of incorrect data bits from 0.1% to 10% and repeat the first example. The results are shown in Fig. 6. We see that DEMA performs reasonably well when only a small number (e.g., 1%) of data bits are in error. As more data bits are estimated incorrectly, the degradation of DEMA becomes more significant. For example, when 5% of the estimated bits are in error, the probability of correct acquisition drops to 0.774.

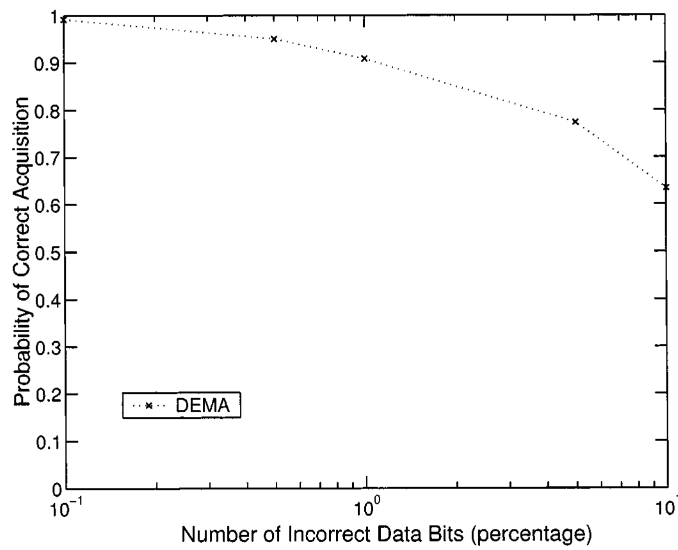

Fig. 6. Probability of correct acquisition versus the percentage of incorrectly estimated data bits when $K = 10, N = 31, Q = 1, M = 100, E_b/N_0 = 10$ dB, and $\xi_k \sim N(10, 100)$.

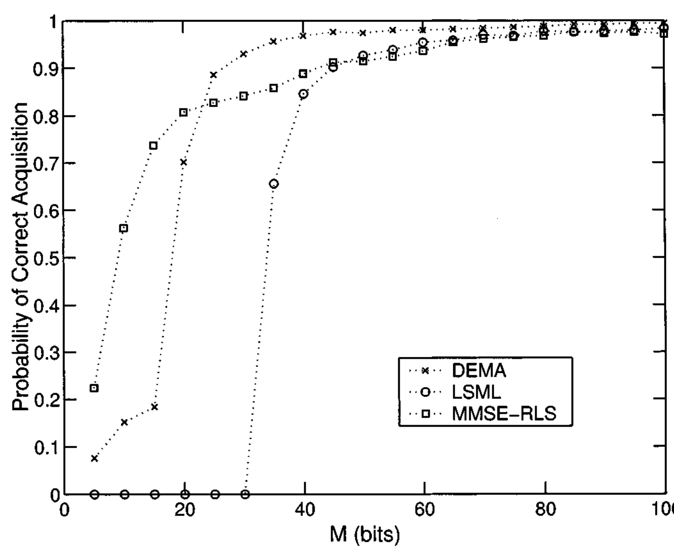

Fig. 7. Probability of correct acquisition versus M when $K = 10, N = 31, Q = 1, M = 100, E_b/N_0 = 10$ dB, $\xi_k \sim N(10, 100)$, and two users transmit identical data sequences.

Correlated Data Bits: The derivation of DEMA uses the assumption that the data bits for all users are i.i.d. Conceivably, if the data bits transmitted by different users are heavily correlated (which, however, seldom happens in practice), the performance of DEMA could degrade considerably. This problem can be solved under certain conditions, as explained in the sequel. In an asynchronous systems where users enter and leave a cell asynchronously, most active users will typically be in normal transmission and only a few new users will need training. Since the number of new users is unlikely to be very large, it is possible to choose a training pattern so that the training bits are

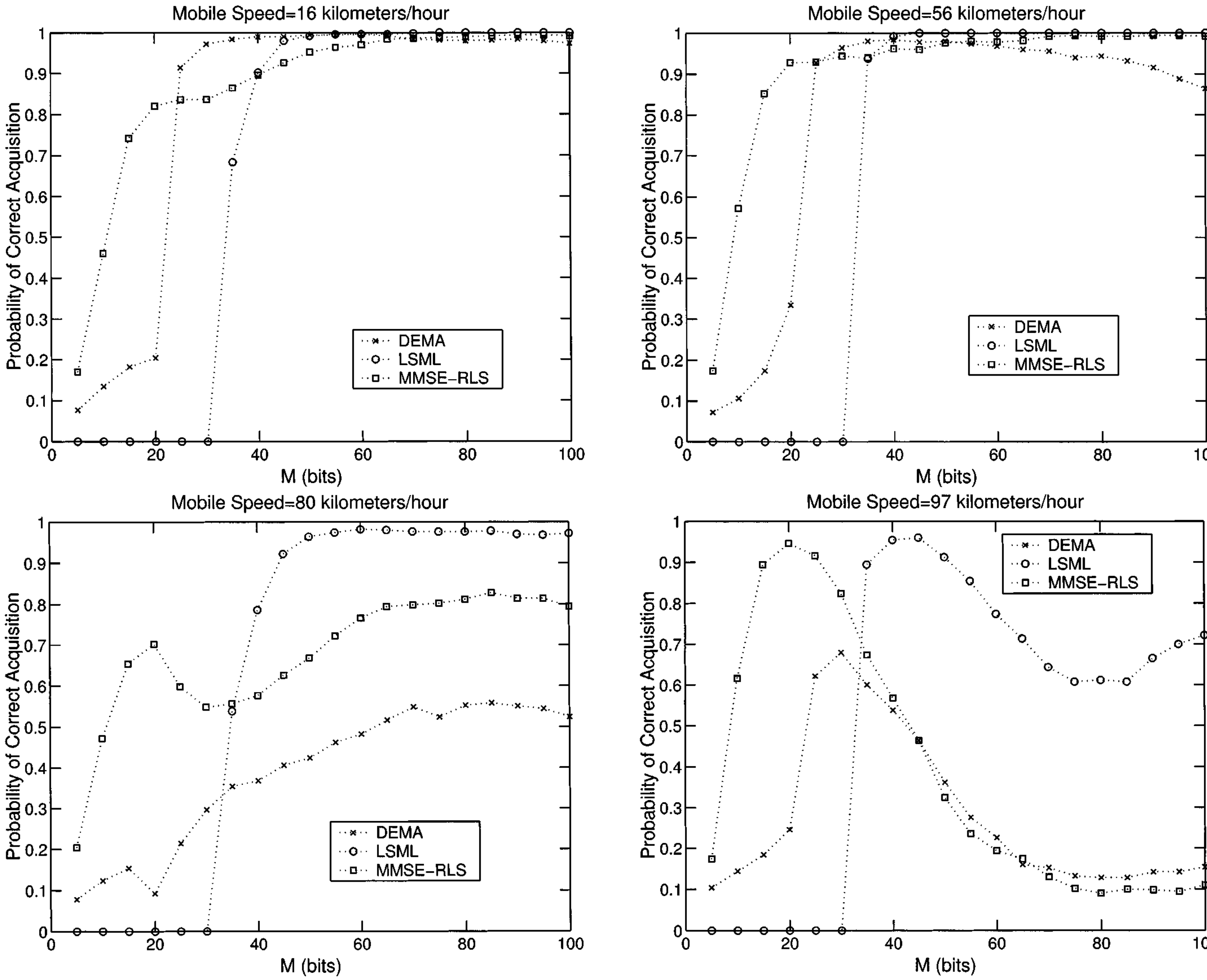

Fig. 8. Probability of correct acquisition versus M when $K = 10, N = 31, Q = 1, E_b/N_0 = 10$ dB, and $\xi_k \sim N(10, 100)$ in a time-varying Rayleigh fading channel. (a) Mobile speed: 16 km/h $(f_D T_s = 0.0013)$. (b) Mobile speed: 56 km/h $(f_D T_s = 0.0047)$. (c) Mobile speed: 80 km/h $(f_D T_s = 0.0067)$. (d) Mobile speed: 97 km/h $(f_D T_s = 0.008)$.

not or moderately correlated. The training bits transmitted by the new users and the data bits transmitted by users in normal transmission may be correlated. The effect of this correlation can be made small by exploiting the SNR consistency property of DEMA (see Remark 1 in Section III) and directing the new users to increase the transmitting power for training. The data bits transmitted by users in normal transmission may or may not be highly correlated. This correlation, however, has little effect on the code-timing estimation for the new users. We show this using an example. The example is similar to the first example except that the data bits for the second and third users are identical (completely correlated) all the time. Since $\mathbf{R}_{\mathbf{ss}}(M)$ is always rank deficient in this case, we use the moore-penrose pseudo-inverse to compute $\mathbf{R}_{\mathbf{ss}}^{-1}(M)$. The results for the first user are shown in Fig. 7. If we compare Figs. 1 and 7, we see that the correlation between the second and third users almost has no effect on the timing estimates for the first user.

Time-Varying Rayleigh Fading Channel: Although all three timing estimators under discussion assume that the channel remains relatively unchanged over the time interval of code acquisition, it would be of interest to see how they perform in a time-varying Rayleigh fading environment. To that end, we repeat the first example by replacing the time-invariant fading coefficients with time-varying ones and keeping the other parameters unchanged. The fading process is simulated by generating zero-mean complex gaussian random process whose spectral density is adjusted based on the Doppler rate according to the Jakes model (see, e.g., [20] for details). A mobile cellular system is simulated, where the carrier frequency is 900 MHz, the data rate is 10 Kb/s, and all mobiles move at a constant speed. Fig. 8(a)–(d) shows the probability of correct acquisition versus M when the mobile speed is 16, 56, 80, and 97 km/h, respectively. The normalized doppler rates, defined as $f_D T_b$, for the four situations are 0.0013, 0.0047, 0.0067, and 0.008, respectively. It is seen that DEMA works quite well for low mobile speeds but degrades considerably as the mobile speed increases. This is not surprising since DEMA relies more on the data model described in Section II than the others. When the model reflects the received data reasonably well, DEMA is able to yield the most accurate timing estimates; otherwise, it may suffer from a considerable performance degradation. On the other hand, by assuming a less structured data model, LSML is less sensitive to model mismatch than DEMA. This advantage, however, is achieved at a loss in estimation accuracy when the channel is approximately time-invariant during code acquisition. It should be noted since the time-invariant assumption made by the three estimators is significantly violated in Fig. 8(c) and (d), increasing the length of data bits may or may not improve the acquisition and a performance fluctuation happens. Note also that when the mobile speed exceeds 50 mi/h, all methods under study degrade significantly and, hence, it is not advisable to use them in fast fading channels.

V. CONCLUSIONS

In this paper, we have investigated the problem of multiuser code-timing estimation in DS-CDMA systems. A new code-timing estimation technique, referred to as the DEMA algorithm, has been presented by assuming that the channel is roughly time-invariant during code acquisition. It has been shown that DEMA 1) is decoupled and computationally efficient; 2) coincides with the ML method and achieves the CRB asymptotically (for large M); and 3) obtains the *exact* parameter estimates in the absence of noise even with finite number of data bits which can be heavily correlated.

Although we have only considered flat fading in the study, DEMA can be straightforwardly extended to frequency selective multipath channels. An analysis similar to that in Section III indicates that the DEMA algorithm will be decoupled between different users, but not between the different paths of the same user. Hence, an L_k-dimensional search over the parameter space will be needed, where L_k is the number of paths for the kth user. Even so, the DEMA algorithm in this case will still be much simpler than the exact ML method, which in general requires a search over a $\sum_{k=1}^{K} 3L_k$-dimensional parameter space.

While DEMA works reasonably well for slow fading channels, it degrades significantly when the channel becomes highly time-varying. In a recent study, the LSML algorithm was extended to the case of using a multiple-antenna-based receiver [26]. By exploiting spatial diversity, the proposed algorithm therein appears to be able to better deal with time-varying fading channels. However, the derivation of that algorithm still assumed a time-invariant channel and it will collapse ultimately when the motion of the mobile is relatively high. Apparently, competitive code-timing acquisition techniques for fast fading channels are yet to be discovered.

APPENDIX
CRAMÉR–RAO BOUND

The CRB is a lower bound on the variance of the estimation error. It indicates the best performance that can be achieved by any unbiased estimator. We derive herein the CRB for the parameter estimates corresponding to the data model in (8). It should be noted that the CRB derived in [18] is based on a different data model, as discussed in Section III-A. Moreover, the CRB in [18] is in general a higher bound than the CRB given below since the former assumes less prior information than the latter.

Let $\mathbf{y} = [\mathbf{y}^{\mathrm{T}}(0) \quad \dots \quad \mathbf{y}^{\mathrm{T}}(M-1)]^{\mathrm{T}} \in \mathbb{C}^{MNQ\times 1}$. Let $\mathbf{z}_k \in \{\pm 1\}^{2M\times 1}$ and $\mathbf{n} \in \mathbb{C}^{MNQ\times 1}$ be similarly formed from $\{\mathbf{z}_k(m)\}_{m=0}^{M-1}$ and $\{\mathbf{n}(m)\}_{m=0}^{M-1}$, respectively. Then, $\mathbf{y}$ can be written as (for notational simplicity, we henceforth drop the dependence of $\mathbf{A}_k(\tau_k)$ on τ_k)

$$\mathbf{y} = \sum_{k=1}^{K} [\mathbf{I}_M \otimes (\beta_k \mathbf{A}_k)]\,\mathbf{z}_k + \mathbf{n} \triangleq \mathbf{w} + \mathbf{n}, \tag{47}$$

where $\otimes$ denotes the Kronecker product [23]. Let $\boldsymbol{\tau} = [\tau_1 \quad \dots \quad \tau_K]^{\mathrm{T}}$, $\boldsymbol{\beta} = [\beta_1 \quad \dots \quad \beta_K]^{\mathrm{T}}$, $\bar{\boldsymbol{\beta}} = \Re(\boldsymbol{\beta})$, and $\tilde{\boldsymbol{\beta}} = \Im(\boldsymbol{\beta})$. We denote the unknown parameters in (47) by

$$\boldsymbol{\eta} = \left[\sigma_n^2 \quad \boldsymbol{\tau}^{\mathrm{T}} \quad \bar{\boldsymbol{\beta}}^{\mathrm{T}} \quad \tilde{\boldsymbol{\beta}}^{\mathrm{T}}\right]^{\mathrm{T}} \triangleq \left[\sigma_n^2 \quad \boldsymbol{\phi}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{(3K+1)\times 1}. \tag{48}$$

Note that $\mathbf{w}$ is solely parameterized by $\boldsymbol{\phi} \in \mathbb{R}^{3K\times 1}$ and will be denoted by $\mathbf{w}(\boldsymbol{\phi})$ hereafter. Next, observe that

$$\boldsymbol{\Sigma}(\sigma^2) \triangleq E\left[\mathbf{n}\mathbf{n}^{\mathrm{H}}\right] = \sigma_n^2 \mathbf{I}_{MNQ}. \tag{49}$$

By using the Slepian–Bangs formula (see, e.g., [22]), the CRB matrix for the problem under study is given elementwise by

$$\begin{aligned}\left[\mathrm{CRB}^{-1}(\boldsymbol{\eta})\right]_{pq} &= \mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}(\sigma_n^2)\frac{\partial\boldsymbol{\Sigma}(\sigma_n^2)}{\partial\eta_p}\boldsymbol{\Sigma}^{-1}(\sigma_n^2)\frac{\partial\boldsymbol{\Sigma}(\sigma_n^2)}{\partial\eta_q}\right] \\ &\quad + 2\Re\left[\frac{\partial\mathbf{w}^{\mathrm{H}}(\boldsymbol{\phi})}{\partial\eta_p}\boldsymbol{\Sigma}^{-1}(\sigma_n^2)\frac{\partial\mathbf{w}(\boldsymbol{\phi})}{\partial\eta_q}\right].\end{aligned} \tag{50}$$

Since σ_n^2 and $\boldsymbol{\phi}$ do not share any common elements, it follows that CRB($\boldsymbol{\eta}$) is block diagonal and the block that corresponds to the signal parameter vector $\boldsymbol{\phi}$ is given by

$$\mathrm{CRB}^{-1}(\boldsymbol{\phi}) = 2\Re\left[\frac{\partial\mathbf{w}^{\mathrm{H}}(\boldsymbol{\phi})}{\partial\boldsymbol{\phi}}\boldsymbol{\Sigma}^{-1}\frac{\partial\mathbf{w}(\boldsymbol{\phi})}{\partial\boldsymbol{\phi}^{\mathrm{T}}}\right]. \tag{51}$$

Next, we evaluate the partial differentiation in (51) as follows:

$$\frac{\partial\mathbf{w}(\boldsymbol{\phi})}{\partial\boldsymbol{\phi}^{\mathrm{T}}} = \left[\mathbf{J}_{\boldsymbol{\tau}} \quad \mathbf{J}_{\boldsymbol{\beta}} \quad j\mathbf{J}_{\boldsymbol{\beta}}\right] \tag{52}$$

where

$$\mathbf{J}_{\boldsymbol{\tau}} = \left[[\mathbf{I}_M \otimes (\beta_1\mathbf{D}_1)]\,\mathbf{z}_1 \quad \dots \quad [\mathbf{I}_M \otimes (\beta_K\mathbf{D}_K)]\,\mathbf{z}_K\right] \tag{53}$$

and

$$\mathbf{J}_{\boldsymbol{\beta}} = \left[(\mathbf{I}_M \otimes \mathbf{A}_1)\mathbf{z}_1 \quad \dots \quad (\mathbf{I}_M \otimes \mathbf{A}_K)\mathbf{z}_K\right]. \tag{54}$$

The matrix $\mathbf{D}_k$ in (53) is given by

$$\mathbf{D}_k = \left[\mathbf{F}_k^{(1)}\mathbf{k} \quad \mathbf{F}_k^{(2)}\mathbf{k}\right] \in \mathbb{R}^{NQ\times 2} \tag{55}$$

where $\mathbf{k} = [-1 \quad 1]^{\mathrm{T}}$. Using (52) in (51) yields

$$\mathrm{CRB}(\boldsymbol{\phi}) = \frac{\sigma_n^2}{2}\left(\Re\begin{bmatrix} \mathbf{J}_{\boldsymbol{\tau}\boldsymbol{\tau}} & \mathbf{J}_{\boldsymbol{\tau}\boldsymbol{\beta}} & j\mathbf{J}_{\boldsymbol{\tau}\boldsymbol{\beta}} \\ \mathbf{J}_{\boldsymbol{\tau}\boldsymbol{\beta}}^{\mathrm{H}} & \mathbf{J}_{\boldsymbol{\beta}\boldsymbol{\beta}} & j\mathbf{J}_{\boldsymbol{\beta}\boldsymbol{\beta}} \\ -j\mathbf{J}_{\boldsymbol{\tau}\boldsymbol{\beta}}^{\mathrm{H}} & -j\mathbf{J}_{\boldsymbol{\beta}\boldsymbol{\beta}} & \mathbf{J}_{\boldsymbol{\beta}\boldsymbol{\beta}} \end{bmatrix}\right)^{-1} \tag{56}$$

where

$$\mathbf{J}_{\boldsymbol{\tau}\boldsymbol{\tau}} = \mathbf{J}_{\boldsymbol{\tau}}^{\mathrm{H}}\mathbf{J}_{\boldsymbol{\tau}} = \sum_{m=0}^{M-1} \mathbf{Z}^{\mathrm{H}}(m)\mathbf{C}^{\mathrm{H}}\mathbf{C}\mathbf{Z}(m) \tag{57}$$

$$\mathbf{J}_{\boldsymbol{\tau}\boldsymbol{\beta}} = \mathbf{J}_{\boldsymbol{\tau}}^{\mathrm{H}}\mathbf{J}_{\boldsymbol{\beta}} = \sum_{m=0}^{M-1} \mathbf{Z}^{\mathrm{H}}(m)\mathbf{C}^{\mathrm{H}}\mathbf{A}\mathbf{Z}(m) \tag{58}$$

and

$$\mathbf{J}_{\boldsymbol{\beta}\boldsymbol{\beta}} = \mathbf{J}_{\boldsymbol{\beta}}^{\mathrm{H}}\mathbf{J}_{\boldsymbol{\beta}} = \sum_{m=0}^{M-1} \mathbf{Z}^{\mathrm{H}}(m)\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{Z}(m). \tag{59}$$

The matrices $\mathbf{A}$, $\mathbf{C}$, and $\mathbf{Z}(m)$ in (57) –(59) are defined as

$$\mathbf{A} = [\,\mathbf{A}_1 \quad \ldots \quad \mathbf{A}_K\,] \in \mathbb{R}^{NQ\times 2K} \tag{60}$$

$$\mathbf{C} = [\,\beta_1\mathbf{D}_1 \quad \ldots \quad \beta_K\mathbf{D}_K\,] \in \mathbb{C}^{NQ\times 2K} \tag{61}$$

and

$$\mathbf{Z}(m) = \begin{bmatrix} \mathbf{z}_1(m) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{z}_K(m) \end{bmatrix} \in \{\pm 1\}^{2K\times K}. \tag{62}$$

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments which led to a significant improvement of the manuscript.

REFERENCES

[1] S. Verdú, "Minimum probability of error for asynchronous gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85–96, Jan. 1986.
[2] R. Lupas and S. Verdú, "Near–far resistance of multiuser detectors in asynchronous channels," *IEEE Trans. Commun.*, vol. 38, pp. 496–508, Apr. 1990.
[3] Z. Xie, R. T. Short, and C. K. Rushforth, "A family of suboptimum detectors for coherent multiuser communications," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 683–690, May 1990.
[4] S. L. Miller, "An adaptive direct-sequence code-division multiple-access receiver for multiuser interference rejection," *IEEE Trans. Commun.*, vol. 43, pp. 1746–1755, Feb./Mar./Apr. 1995.
[5] H. V. Poor, "On parameter estimation in DS/CDMA formats," in *Advances in Communications and Signal Processing*, W. A. Porter and S. C. Kak, Eds. New York: Springer-Verlag, 1989.
[6] Z. Xie, C. K. Rushforth, R. T. Short, and T. K. Moon, "Joint signal detection and parameter estimation in multiuser communications," *IEEE Trans. Commun.*, vol. 41, pp. 1208–1215, Aug. 1993.
[7] K. K. Chawla and D. C. Sarwate, "Parallel acquisition of PN sequences in DS/SS systems," *IEEE Trans. Commun.*, vol. 42, pp. 2155–2164, May 1994.
[8] A. M. Slonneger and D. V. Sarwate, "Noncoherent parallel acquisition of PN sequences in direct-sequence spread-spectrum systems," in *Proc. IEEE 2nd Int. Symp. Spread Spectrum Techniques Applications*, 1992, pp. 31–34.
[9] R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to Spread Spectrum Communications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[10] R. F. Smith and S. L. Miller, "Code timing estimation in a near–far environment for direct-sequence code-division multiple-access," in *Proc. 1994 IEEE Military Communication Conf.*, 1994, pp. 47–51.
[11] E. G. Ström, S. Parkvall, 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.
[12] S. E. Bensley and B. Aazhang, "Subspace-based channel estimation for code division multiple access communications systems," *IEEE Trans. Commun.*, vol. 44, pp. 1009–1020, Aug. 1996.
[13] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagat.*, vol. AP-34, pp. 276–280, Mar. 1986.
[14] P. Zhang, R. T. Short, and X. X. Chen, "Fast timing estimation for CDMA waveforms in a near–far environment," in *Proc. IEEE GLOBECOM*, Nov. 1996, pp. 1799–1803.
[15] C.-M. Chang and K.-C. Chen, "A new interpretation to DS-CDMA multiuser synchronization," in *Proc. IEEE PIMRC*, Sept. 1999.
[16] ——, "Joint linear timing and carrier phase estimation of DS-CDMA multiuser communications," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 87–98, Jan. 2000.
[17] J. Li, B. Halder, P. Stoica, and M. Viberg, "Computationally efficient angle estimation for signals with known waveforms," *IEEE Trans. Signal Processing*, vol. 43, pp. 2154–2163, Sept. 1995.
[18] D. Zheng, J. Li, S. L. Miller, and E. G. Ström, "An efficient code-timing estimator for DS-CDMA system," *IEEE Trans. Signal Processing*, vol. 45, pp. 82–89, Jan. 1997.
[19] S. E. Bensley and B. Aazhang, "Maximum-likelihood synchronization of a single user for code-division multiple-access communication systems," *IEEE Trans. Commun.*, vol. 46, pp. 392–399, Mar. 1998.
[20] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[21] R. A. Iltis, "Performance of constrained and unconstrained adaptive multiuser detectors for quasisynchronous CDMA," *IEEE Trans. Commun.*, vol. 46, pp. 135–143, Jan. 1998.
[22] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
[24] T. Ostman and B. Ottersten, "Near far robust time delay estimation for asynchronous DS-CDMA systems with bandlimited pulse shapes," in *Proc. IEEE 48th Vehicular Technology Conf.*, Ottawa, ON, Canada, May 1998, pp. 1650–1654.
[25] U. Madhow, "Adaptive interference suppression for joint acquisition and demodulation of direct-sequence CDMA signals," in *Proc. 1995 IEEE Military Communication Conf.*, 1995, pp. 1200–1204.
[26] Z.-S. Liu, J. Li, and S. L. Miller, "An efficient code-timing estimator for receiver diversity DS-CDMA systems," *IEEE Trans. Commun.*, vol. 46, pp. 826–835, June 1998.

Hongbin Li (S'98–M'99) received the B.S. and M.S. degrees from the University of Electronic Science and Technology, China, in 1991 and 1994, respectively, and the Ph.D. degree from the University of Florida, Gainesville, in 1999, all in electrical engineering.

From July 1996 to May 1999, he was a Research Assistant in the Department of Electrical and Computer Engineering at the University of Florida, Gainesville. Since July 1999, he has been an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ. His current research interests include wireless communications, statistical signal processing, sensor array processing, and radar imaging.

Dr. Li is a member of Tau Beta Pi and Phi Kappa Phi. He received the 1999 Sigma Xi Graduate Research Award.

Jian Li (S'87–M'91–SM'97) received the M.Sc. and Ph.D. degrees in electrical engineering from The Ohio State University, Columbus, in 1987 and 1991, respectively.

From April 1991 to June 1991, she was an Adjunct Assistant Professor with the Department of Electrical Engineering, The Ohio State University, Columbus. From July 1991 to June 1993, she was an Assistant Professor with the Department of Electrical Engineering, University of Kentucky, Lexington. Since August 1993, she has been with the Departmentof Electrical and Computer Engineering, University of Florida, Gainesville, where she is currently a Professor.Her current research interests include spectral estimation, array signal processing, and signal processing for wireless communications and radar.

Dr. Li is a member of Sigma Xi and Phi Kappa Phi. She is currently an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING and a Guest Editor for *Multidimensional Systems and Signal Processing*. She received the 1994 National Science Foundation Young Investigator Award and the 1996 Office of Naval Research Young Investigator Award.

Scott L. Miller (S'87–M'88–SM'97) was born in Los Angeles, CA, in 1963. He received the B.S., M. S., and Ph. D. degrees in electrical engineering from the University of California at San Diego (UCSD) in 1985, 1986, and 1988, respectively.

He then joined the Department of Electrical and Computer Engineering at the University of Florida, where he was an Assistant Professor from 1988 through 1993 and an Associate Professor from 1993 through 1998. In August 1998, he joined the Electrical Engineering Department at Texas A&M University. He has also held visiting positions at Motorola Inc., University of Utah, and UCSD. He has taught courses at both the graduate and undergraduate level on such topics as signals and systems, engineering mathematics, digital and analog communications, probability and random processes, coding, information theory, spread spectrum, detection and estimation theory, wireless communications, queuing theory and communication networks. He has published over 75 refereed journal and conference papers on a variety of topics in the area of digital communication theory. His current research interests are in the area of wireless communications with a special emphasis on CDMA systems.

Dr. Miller is an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.