EURASIP Journal on Applied Signal Processing 2005:5, 670–682

An Efficient Code-Timing Estimator for DS-CDMA Systems over Resolvable Multipath Channels

Jianhua Liu

College of Engineering, Embry-Riddle Aeronautical University, 600 S. Clyde Morris Boulevard, Daytona Beach, FL 32114, USA
Email: jianhua.liu@erau.edu

Jian Li

Department of Electrical and Computer Engineering, University of Florida, P.O. Box 116130, Gainesville, FL 32611-6130, USA
Email: li@dsp.ufl.edu

Received 31 July 2003; Revised 26 March 2004

We consider the problem of training-based code-timing estimation for the asynchronous direct-sequence code-division multiple-access (DS-CDMA) system. We propose a modified large-sample maximum-likelihood (MLSML) estimator that can be used for the code-timing estimation for the DS-CDMA systems over the resolvable multipath channels in closed form. Simulation results show that MLSML can be used to provide a high correct acquisition probability and a high estimation accuracy. Simulation results also show that MLSML can have very good near-far resistant capability due to employing a data model similar to that for adaptive array processing where strong interferences can be suppressed.

Keywords and phrases: DS-CDMA, synchronization, code timing, multipath channels.

1. INTRODUCTION

Direct-sequence code-division multiple access (DS-CDMA) is one of the most promising multiple-access technologies for the next-generation wireless communication services. For time-dispersive fading channels, DS-CDMA can outperform other multiple-access schemes, such as FDMA (frequency-division multiple access) and TDMA (time-division multiple access), due to its capability of exploiting the RAKE combination [1] to combat the time-dispersive fading problem effectively.

The structure of the RAKE receiver is determined by the type of the time-dispersive multipath channel over which the DS-CDMA system works. There are mainly two kinds of time-dispersive multipath channels for the DS-CDMA systems. The first one is the (nearly) continuous channel, which can be represented by a finite impulse response (FIR) filter channel model; the second one is the discrete channel where the separations of the resolvable bunches of multipaths (referred to as resolvable paths in the sequel) are larger than the chip duration of the spreading code. The latter can be represented by a resolvable time-dispersive channel model. While the first model is quite effective in describing channels in urban areas, the second model is suitable for describing channels in rural or mountain areas or in the case of soft handoff [2], where a few resolvable paths exist. In this paper, we focus on the channels represented by the second model and we refer to them as the *resolvable multipath channels*.

For the resolvable multipath channels considered herein, the RAKE receiver assumes the knowledge of the channel parameters including the code timing, signal power, as well as the carrier phase for each resolvable path of each user. These parameters are typically unknown in practice and hence need to be estimated. In this paper, we consider the problem of code-timing estimation for the asynchronous DS-CDMA system. Given an accurate code-timing estimate, there is a wealth of good methods for estimating the other parameters [3].

A standard technique for code-timing estimation is the training-based correlator (or matched filter) [4] which needs a known training sequence. It is well known that the correlator coincides with the optimal maximum-likelihood (ML) method in the single-user case for the Gaussian channel. The correlator can also be used for the resolvable multipath channels in the single-user case. However, in the multiuser case, its performance degrades considerably because of the so-called near-far problem.

A near-far resistant code-timing estimator is the MUSIC algorithm, which was proposed independently in [5, 6]. Unlike the aforementioned correlator, the MUSIC algorithm is a blind approach, that is, it needs no training for the purpose

of code-timing estimation. Hence, the MUSIC algorithm can be used to perform not only code-timing estimation (acquisition), but also tracking. Yet MUSIC suffers from low correct acquisition probability and poor estimation accuracy, as well as poor subscriber capacity.

Another near-far resistant code-timing estimator, referred to as the large-sample ML (LSML) algorithm, was proposed in [7]. LSML is also training-based and can be used to perform code-timing estimation for the asynchronous DS-CDMA system over time-invariant *flat fading* channels. Unlike conventional ML estimators, which employ computationally extensive multidimensional search, LSML uses a closed-form solution, which is computationally very efficient. The underlying idea of LSML was extended in [8] to derive a training-based ML method to estimate the code timing for resolvable multipath channels. However, the method in [8] needs to perform a multidimensional search to obtain the code-timing estimation for each resolvable path, which is computationally very heavy.

In this paper, we propose a new training-based code-timing estimation method, referred to as the modified LSML (MLSML) algorithm. The MLSML algorithm can be used to avoid the multidimensional search problem of code-timing estimation of [8] by using a closed-form solution similar to LSML and hence is computationally very efficient. Moreover, we show that the MLSML algorithm can be used in the case of arbitrary resolvable delays, that is, the resolvable delays can be in different processing windows and therefore relax the constraints in [7, 8]. As a result, the MLSML algorithm is easier to use in the practical multipath channel scenarios. Simulations show that MLSML can be used to provide a high correct acquisition probability and a high estimation accuracy. Simulations also show that MLSML can have very good near-far resistant capability, due to employing a data model similar to that for adaptive array processing where strong interferences can be suppressed.

The remainder of this paper is organized as follows. In Section 2, we provide the data model for code-timing estimation. In Section 3, we derive the MLSML algorithm in detail. Numerical examples are presented in Section 4 to demonstrate the effectiveness of the proposed MLSML algorithm. Finally, we end our paper with comments and conclusions in Section 5.

2. DATA MODEL

Consider an asynchronous DS-CDMA system employing BPSK modulation (it can be readily generalized to other PSK modulations), operating over time-invariant resolvable multipath channels. We assume that all the users use the same carrier frequency and there is no frequency offset between the transmitters and the receivers. We consider the case of using short codes. The spreading code for each user is a pseudonoise (PN) sequence with length N.

Let

$$c_k(t) = \sum_{n=0}^{N-1} c_k(n) P_{T_C}(t - nT_C) \tag{1}$$

be one period of the spreading waveform for user k, which has a period of $T = NT_C$, where T is the duration of the information carrying data symbol, T_C is the chip duration, and $P_{T_C}(t)$ denotes the unit rectangular pulse on $[0, T_C)$. The kth user's transmitted signal has the form

$$x_k(t) = \sqrt{2P_k}\, s_k(t) \cos(\omega_0 t + \theta_k), \tag{2}$$

where P_k is the transmission power of user k, ω_0 is the common carrier frequency, θ_k is the carrier phase, and

$$s_k(t) = \sum_{m=0}^{M} d_k(m) c_k(t - mT) \tag{3}$$

is the baseband spread spectrum signal with $d_k(m) \in \{-1, +1\}$, $m = 0, 1, \ldots, M$, being the known training data symbol.

The resolvable multipath channel for user k can be described as

$$h_k(t) = \sum_{l=1}^{L_k} \alpha_{k,l} \delta(t - \tau_{k,l}), \tag{4}$$

where $\delta(t)$ is the Dirac delta, $\alpha_{k,l}$ is the complex-valued fading coefficient of the lth path, and $\tau_{k,l}$ is the time delay between the transmitter and the receiver on the lth path with $\tau_{k,l+1} - \tau_{k,l} \geq T_C$, $l = 1, 2, \ldots, L_k - 1$. The noise-free received baseband analog signal can be written as

$$r_A(t) = \sum_{k=1}^{K} \sum_{l=1}^{L_k} \beta_{k,l} s_k(t - \tau_{k,l}), \tag{5}$$

where

$$\beta_{k,l} \triangleq \sqrt{P_k}\, \alpha_{k,l} e^{j(\theta_k - \omega_0 \tau_{k,l})}. \tag{6}$$

We employ a receiver with the front end consisting of an in-phase/quadrature (I/Q) mixer followed by an integrate-and-dump filter with integration time T_C. Then, the noise-free output digital sequence for the analog signal is

$$r(i) = \frac{1}{T_C} \sum_{k=1}^{K} \sum_{l=1}^{L_k} \beta_{k,l} \int_{(i-1)T_C}^{iT_C} s_k(t - \tau_{k,l})\, dt. \tag{7}$$

We consider the case that $\tau_{k,L_k} - \tau_{k,1} < (N-1)T_C$. It is assumed in [7, 8] that all the time delays are within $[0, T)$, that is, $\tau_{k,l} \in [0, T)$ for all k and l. While this assumption can be reasonable for the flat fading case, it is no longer valid in the case of resolvable multipath channels. The reason is that if the delay of the first path is within $[0, T)$, the paths with longer delays can be within $[T, 2T)$. Hence we assume that the time delays are within $[0, 2T)$ in this paper. For notational simplicity, in the derivation of the data model, we first assume that the time delays are within $[0, T)$. Then we will address the case that the time delays are within $[0, 2T)$ later on.

For $\tau_{k,l} \in [0, T)$, we consider it as the summation of multiple chip durations and a residue, that is, let $\tau_{k,l} = p_{k,l}T_C + \gamma_{k,l}$, where $p_{k,l} \in \{0, 1, \ldots, N-1\}$ and $\gamma_{k,l} \in [0, T_C)$. The (k,l)th term of the integration in (7) is then given by

$$\begin{aligned} r_{k,l}(i) &\triangleq \frac{1}{T_C}\int_{(i-1)T_C}^{iT_C} s_k(t-\tau_{k,l})\,dt \\ &= \frac{1}{T_C}\sum_{m=0}^{M} d_k(m)\sum_{n=0}^{N-1} c_k(n) \\ &\quad\times \int_{(i-1)T_C}^{iT_C} P_{T_C}\big(t-(mN+n+p_{k,l})T_C-\gamma_{k,l}\big)\,dt \\ &= \frac{T_C-\gamma_{k,l}}{T_C}\sum_{m=0}^{M} d_k(m)\sum_{n=0}^{N-1} c_k(n)\delta\big(i-(mN+n+p_{k,l})-1\big) \\ &\quad+\frac{\gamma_{k,l}}{T_C}\sum_{m=0}^{M} d_k(m)\sum_{n=0}^{N-1} c_k(n)\delta\big(i-(mN+n+p_{k,l})-2\big) \\ &= \Big(1-\frac{\gamma_{k,l}}{T_C}\Big)c_k(i-m_1N-p_{k,l}-1)\,d_k(m_1) \\ &\quad+\frac{\gamma_{k,l}}{T_C}c_k(i-m_2N-p_{k,l}-2)\,d_k(m_2), \end{aligned} \tag{8}$$

which is a function of $p_{k,l}$ and $\gamma_{k,l}$, the parameters to be estimated. Here m_1 and m_2 are the indexes for the training data symbols such that $0 \le i - m_1N - p_{k,l} - 1 \le N-1$ and $0 \le i - m_2N - p_{k,l} - 2 \le N-1$, respectively. In what follows, we determine m_1 and m_2 for each i.

From (8) we know that for $i - p_{k,l} - 2 = mN + N - 1$, that is, $i = (m+1)N + p_{k,l} + 1$, we have $m_2 = m$ and $m_1 = m+1$, which leads to

$$r_{k,l}(mN+N+p_{k,l}+1) = \nu c_k(0)d_k(m+1) + \mu c_k(N-1)d_k(m), \tag{9}$$

where $\mu \triangleq \gamma_{k,l}/T_C$ and $\nu \triangleq 1-\mu$. Similarly for $i = mN+N+1, \ldots, mN+N+p_{k,l}$, we have

$$r_{k,l}(i) = \big[\nu c_k(i-mN-p_{k,l}-1) + \mu c_k(i-mN-p_{k,l}-2)\big]d_k(m), \tag{10}$$

and for $i = mN+N+p_{k,l}+2, \ldots, mN+2N$, we have

$$\begin{aligned} r_{k,l}(i) = \big[&\nu c_k(i-mN-N-p_{k,l}-1) \\ &+\mu c_k(i-mN-N-p_{k,l}-2)\big]d_k(m+1). \end{aligned} \tag{11}$$

Stacking $r_{k,l}$'s of (9), (11), and (10) in a vector, we have (by letting $p = p_{k,l}$ for notational convenience)

$$\begin{aligned} \mathbf{r}_{k,l}(m+1) &\triangleq \begin{bmatrix} r_{k,l}(mN+2N) \\ \vdots \\ r_{k,l}(mM+N+1) \end{bmatrix} \\ &= \big[\nu\mathbf{J}_1(p)\mathbf{c}_k + \mu\mathbf{J}_1(p+1)\mathbf{c}_k \quad \nu\mathbf{J}_2(p)\mathbf{c}_k + \mu\mathbf{J}_2(p+1)\mathbf{c}_k\big] \\ &\quad\times \mathbf{d}_k(m+1), \end{aligned} \tag{12}$$

where

$$\mathbf{c}_k \triangleq \big[c_k(N-1) \quad c_k(N-2) \quad \cdots \quad c_k(0)\big]^T \tag{13}$$

with $(\cdot)^T$ denoting the transpose,

$$\mathbf{J}_1(p) \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_p & \mathbf{0} \end{bmatrix}, \qquad \mathbf{J}_2(p) \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I}_{N-p} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{14}$$

with $\mathbf{I}_p$ being the $p \times p$ identity matrix and $\mathbf{0}$ being a zero matrix with required dimensions, and

$$\mathbf{d}_k(m+1) \triangleq \big[d_k(m) \quad d_k(m+1)\big]^T. \tag{15}$$

By denoting

$$\mathbf{z}_k(m+1) \triangleq \frac{1}{2}\begin{bmatrix} d_k(m+1) + d_k(m) \\ d_k(m+1) - d_k(m) \end{bmatrix}, \tag{16}$$

we have

$$\mathbf{r}_{k,l}(m+1) = \big[\mathbf{a}_1(\tau_{k,l}) \quad \mathbf{a}_2(\tau_{k,l})\big]\mathbf{z}_k(m+1), \tag{17}$$

where

$$\begin{aligned} \mathbf{a}_1(\tau_{k,l}) &\triangleq \{\mu[\mathbf{J}_2(p+1) + \mathbf{J}_1(p+1)] + \nu[\mathbf{J}_2(p) + \mathbf{J}_1(p)]\}\mathbf{c}_k, \\ \mathbf{a}_2(\tau_{k,l}) &\triangleq \{\mu[\mathbf{J}_2(p+1) - \mathbf{J}_1(p+1)] + \nu[\mathbf{J}_2(p) - \mathbf{J}_1(p)]\}\mathbf{c}_k. \end{aligned} \tag{18}$$

Hence, we have

$$\begin{aligned} \mathbf{r}_k(m+1) &\triangleq \sum_{l=1}^{L_k} \mathbf{r}_{k,l}(m+1) \\ &= \sum_{l=1}^{L_k} \beta_{k,l}\big[\mathbf{a}_1(\tau_{k,l}) \quad \mathbf{a}_2(\tau_{k,l})\big]\mathbf{z}_k(m+1) \\ &= \big[\mathbf{A}_{k,1}\boldsymbol{\beta}_k \quad \mathbf{A}_{k,2}\boldsymbol{\beta}_k\big]\mathbf{z}_k(m+1), \end{aligned} \tag{19}$$

where

$$\begin{aligned} \mathbf{A}_{k,1} &\triangleq \big[\mathbf{a}_1(\tau_{k,1}) \quad \mathbf{a}_1(\tau_{k,2}) \quad \cdots \quad \mathbf{a}_1(\tau_{k,L_k})\big], \\ \mathbf{A}_{k,2} &\triangleq \big[\mathbf{a}_2(\tau_{k,1}) \quad \mathbf{a}_2(\tau_{k,2}) \quad \cdots \quad \mathbf{a}_2(\tau_{k,L_k})\big], \\ \boldsymbol{\beta}_k &\triangleq \big[\beta_{k,1} \quad \beta_{k,2} \quad \cdots \quad \beta_{k,L_k}\big]^T. \end{aligned} \tag{20}$$

Note that instead of $\mathbf{z}_k(m+1)$ in (17) we could have used $\mathbf{d}_k(m+1) = [d_k(m) \quad d_k(m+1)]^T$. Yet, the autocorrelation of the former,

$$\begin{aligned} &\mathrm{E}\big[\mathbf{d}_k(m+1)\mathbf{d}_k^H(m)\big] \\ &\quad= \frac{1}{4}\begin{bmatrix} \mathrm{E}[d_k(m)d_k^*(m-1)] & \mathrm{E}\big|d_k(m)\big|^2 \\ \mathrm{E}[d_k(m+1)d_k^*(m-1)] & \mathrm{E}[d_k(m+1)d_k^*(m)] \end{bmatrix}, \\ &\quad= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{21}$$

is not as scattered as that of the latter,

$$\begin{aligned}\mathrm{E}[\mathbf{z}_k(m+1)\mathbf{z}_k^H(m)] &= \frac{1}{4}\begin{bmatrix} \mathrm{E}\,|d_k(m)|^2 & \mathrm{E}\,|d_k(m)|^2 \\ -\mathrm{E}\,|d_k(m)|^2 & -\mathrm{E}\,|d_k(m)|^2 \end{bmatrix} \\ &= \frac{1}{4}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix},\end{aligned} \tag{22}$$

which is being preferred.

By expressing the multiple-access interference (MAI) in a similar way and lumping the MAI with the additive noise into $\mathbf{w}(m)$, we have the data model for the kth user as

$$\mathbf{r}(m) = \widetilde{\mathbf{D}}_k\mathbf{z}_k(m) + \mathbf{w}(m), \quad m = 1, 2, \ldots, M, \tag{23}$$

where

$$\widetilde{\mathbf{D}}_k \triangleq [\mathbf{A}_{k,1}\boldsymbol{\beta}_k \;\; \mathbf{A}_{k,2}\boldsymbol{\beta}_k]. \tag{24}$$

The vector $\mathbf{w}(m)$ is assumed to be independent of the desired signal and to be a white circularly symmetric complex Gaussian random vector with zero-mean and covariance matrix $\mathbf{Q}$.

The problem of interest herein is to estimate $\tau_{k,l}$, $l = 1, 2, \ldots, L_k$, of the interested kth user from $\mathbf{r}(m)$ based on the assumptions summarized below:

(A1) L_k is known and $\tau_{k,l+1} - \tau_{k,l} \geq T_C$ for $l = 1, 2, \ldots, L_k - 1$ with $\tau_{k,L_k} - \tau_{k,1} < (N-1)T_C$;
(A2) $\mathbf{z}_k(m)$, $m = 1, 2, \ldots, M$, is known;
(A3) $\mathbf{c}_k$ is known and has low autocorrelation;
(A4) $\mathbf{Q}$ is arbitrary—determined by other users and noise.

3. THE MLSML ALGORITHM

We present the MLSML algorithm in this section. For notational convenience, we will drop the notational dependence on k in the sequel.

The log-likelihood function of the receiver output vector $\mathbf{r}(m)$, $m = 1, 2, \ldots, M$, is proportional to

$$F = -\ln|\mathbf{Q}| - \mathrm{tr}\left\{\mathbf{Q}^{-1}\frac{1}{M}\sum_{m=1}^{M}[\mathbf{r}(m) - \mathbf{D}\mathbf{z}(m)][\mathbf{r}(m) - \mathbf{D}\mathbf{z}(m)]^H\right\}, \tag{25}$$

which is a function of unknown $\mathbf{Q}$ and $\mathbf{D}_k$, with the latter being determined by unknown code-timing and complex amplitude (cf. (6), (20), and (24)). Here $|\cdot|$ denotes the determinant of a matrix and $(\cdot)^H$ denotes the conjugate transpose. The focus of this paper is on estimating the unknown code-timing by minimizing the cost function F.

By taking the derivative [9], we have

$$\frac{\partial F}{\partial Q_i} = -\mathrm{tr}\left\{\mathbf{Q}^{-1}\frac{\partial \mathbf{Q}}{\partial Q_i}\right\} + \mathrm{tr}\left\{\mathbf{Q}^{-1}\frac{\partial \mathbf{Q}}{\partial Q_i}\mathbf{Q}^{-1}\frac{1}{M}\sum_{m=1}^{M}[\mathbf{r}(m) - \mathbf{D}\mathbf{z}(m)] \times [\mathbf{r}(m) - \mathbf{D}\mathbf{z}(m)]^H\right\}, \tag{26}$$

where Q_i is an element of $\mathbf{Q}$. The above equation means that the cost function of (25) is maximized by

$$\hat{\mathbf{Q}} = \frac{1}{M}\sum_{m=1}^{M}[\mathbf{r}(m) - \mathbf{D}\mathbf{z}(m)][\mathbf{r}(m) - \mathbf{D}\mathbf{z}(m)]^H, \tag{27}$$

which leads to the following cost function to be minimized:

$$F_1 \triangleq \left|\frac{1}{M}\sum_{m=1}^{M}[\mathbf{r}(m) - \mathbf{D}\mathbf{z}(m)][\mathbf{r}(m) - \mathbf{D}\mathbf{z}(m)]^H\right|. \tag{28}$$

An unstructured estimate of $\mathbf{D}$ is obtained as [10]

$$\hat{\mathbf{D}} = \hat{\mathbf{R}}_{zr}^H\hat{\mathbf{R}}_{zz}^{-1}, \tag{29}$$

where

$$\begin{aligned}\hat{\mathbf{R}}_{zr} &\triangleq \frac{1}{M}\sum_{m=1}^{M}\mathbf{z}(m)\mathbf{r}^H(m), \\ \hat{\mathbf{R}}_{zz} &\triangleq \frac{1}{M}\sum_{m=1}^{M}\mathbf{z}(m)\mathbf{z}^H(m).\end{aligned} \tag{30}$$

By plugging (29) into (27), $\hat{\mathbf{Q}}$ can be rewritten as

$$\hat{\mathbf{Q}} = \hat{\mathbf{R}}_{rr} - \hat{\mathbf{R}}_{zr}^H\hat{\mathbf{R}}_{zz}^{-1}\hat{\mathbf{R}}_{zr}, \tag{31}$$

where

$$\hat{\mathbf{R}}_{rr} \triangleq \frac{1}{M}\sum_{m=1}^{M}\mathbf{r}(m)\mathbf{r}^H(m). \tag{32}$$

It has been shown in [10] that minimizing F_1 is asymptotically (for large M) equivalent to minimizing

$$F_2 \triangleq \mathrm{tr}\left[\mathbf{R}_{zz}(\mathbf{D} - \hat{\mathbf{D}})^H\hat{\mathbf{Q}}^{-1}(\mathbf{D} - \hat{\mathbf{D}})\right]. \tag{33}$$

Note that $\mathbf{R}_{zz}$ is a diagonal matrix with equal diagonal elements due to the fact that $d(m)$ can be selected to be an independently and identically distributed sequence and the fact that

$$\mathrm{E}[z_1(m)z_2^*(m)] = \frac{1}{4}\mathrm{E}\left[|d(m+1)|^2 - |d(m)|^2\right] = 0, \tag{34}$$

where $(\cdot)^*$ stands for the complex conjugate. Then, minimizing (33) is equivalent to minimizing the following cost function:

$$F_3 \triangleq [\hat{\mathbf{d}}_1 - \mathbf{A}_1\boldsymbol{\beta}]^H \hat{\mathbf{Q}}^{-1} [\hat{\mathbf{d}}_1 - \mathbf{A}_1\boldsymbol{\beta}] + [\hat{\mathbf{d}}_2 - \mathbf{A}_2\boldsymbol{\beta}]^H \hat{\mathbf{Q}}^{-1} [\hat{\mathbf{d}}_2 - \mathbf{A}_2\boldsymbol{\beta}], \tag{35}$$

where $\hat{\mathbf{d}}_1$ and $\hat{\mathbf{d}}_2$ denote the first and the second columns of $\hat{\mathbf{D}}$, respectively.

A first glance at (35) seems that we need a multidimensional search, as in [8], to estimate τ_l, $l = 1, 2, \ldots, L$. Now, we show that by utilizing the property of low autocorrelation of the spreading code $\mathbf{c}$, we can decouple the multidimensional search into L one-dimensional searches. Furthermore, we show that these L one-dimensional searches can be combined into one one-dimensional search, which can be solved in a closed form.

For notational simplicity, we only consider the first part of (35) in the following derivation. We have

$$\begin{aligned} F_4 &\triangleq \left[\hat{\mathbf{d}}_1 - \sum_{l=1}^{L} \beta_l \mathbf{a}_1(\tau_l)\right]^H \hat{\mathbf{Q}}^{-1} \left[\hat{\mathbf{d}}_1 - \sum_{l=1}^{L} \beta_l \mathbf{a}_1(\tau_l)\right] \\ &= \hat{\mathbf{d}}_1^H \hat{\mathbf{Q}}^{-1} \hat{\mathbf{d}}_1 - 2\,\mathrm{Re}\left\{\hat{\mathbf{d}}_1^H \hat{\mathbf{Q}}^{-1} \sum_{l=1}^{L} \beta_l \mathbf{a}_1(\tau_l)\right\} \\ &\quad + \sum_{l=1}^{L} |\beta_l|^2 \mathbf{a}_1^H(\tau_l) \hat{\mathbf{Q}}^{-1} \mathbf{a}_1(\tau_l) \\ &\quad + \sum_{l_1=1}^{L} \sum_{l_2=1, l_2 \neq l_1}^{L} \beta_{l_1}^* \beta_{l_2} \mathbf{a}_1^H(\tau_{l_1}) \hat{\mathbf{Q}}^{-1} \mathbf{a}_1(\tau_{l_2}). \end{aligned} \tag{36}$$

For $\mathbf{c}$ with low autocorrelation, we have

$$\mathbf{a}_1^H(\tau_{l_1}) \mathbf{a}_1(\tau_{l_2}) = \begin{cases} N, & \tau_{l_1} = \tau_{l_2}, \\ O(1), & |\tau_{l_1} - \tau_{l_2}| \geq T_C. \end{cases} \tag{37}$$

This low autocorrelation property of $\mathbf{c}$ can be exploited to approximately represent the "weighted" autocorrelation as

$$\mathbf{a}_1^H(\tau_{l_1}) \hat{\mathbf{Q}}^{-1} \mathbf{a}_1(\tau_{l_2}) \propto \begin{cases} N, & \tau_{l_1} = \tau_{l_2}, \\ O(1), & |\tau_{l_1} - \tau_{l_2}| \geq T_C, \end{cases} \tag{38}$$

for arbitrary "weighting" matrix $\mathbf{Q}$. Note that the "weighted" autocorrelation of (38) is not as low as the autocorrelation of (37); yet, under the assumption (A1), the former can still be used to yield the following simplified cost function, by dropping the cross items in (36):

$$\begin{aligned} F_4 &\simeq \check{\mathbf{d}}_1^H \check{\mathbf{d}}_1 - 2\,\mathrm{Re}\left\{\check{\mathbf{d}}_1^H \sum_{l=1}^{L} \beta_l \check{\mathbf{a}}_1(\tau_l)\right\} + \sum_{l=1}^{L} |\beta_l|^2 \check{\mathbf{a}}_1^H(\tau_l) \check{\mathbf{a}}_1(\tau_l) \\ &= \sum_{l=1}^{L} [\check{\mathbf{d}}_1 - \beta_l \check{\mathbf{a}}_1(\tau_l)]^H [\check{\mathbf{d}}_1 - \beta_l \check{\mathbf{a}}_1(\tau_l)] - (L-1) \check{\mathbf{d}}_1^H \check{\mathbf{d}}_1 \\ &= \sum_{l=1}^{L} ||\check{\mathbf{d}}_1 - \beta_l \check{\mathbf{a}}_1(\tau_l)||^2 - (L-1) ||\check{\mathbf{d}}_1||^2, \end{aligned} \tag{39}$$

where

$$\check{\mathbf{d}}_1 \triangleq \hat{\mathbf{Q}}^{-1/2} \hat{\mathbf{d}}_1, \qquad \check{\mathbf{a}}_1(\tau_l) \triangleq \hat{\mathbf{Q}}^{-1/2} \mathbf{a}_1(\tau_l), \tag{40}$$

and $\|\cdot\|$ denotes the Euclidean norm. Using the same technique and notation for the second part of (35), we have

$$F_3 \simeq \sum_{l=1}^{L} \left[||\check{\mathbf{d}}_1 - \beta_l \check{\mathbf{a}}_1(\tau_l)||^2 + ||\check{\mathbf{d}}_l - \beta_2 \check{\mathbf{a}}_2(\tau_l)||^2 \right] + \text{const.} \tag{41}$$

Note that (41) is actually a decoupled cost function, which means that we can obtain the code-timing estimates in L one-dimensional searches. That is, for $l = 1, 2, \ldots, L$, we have

$$[\hat{\tau}_l, \hat{\beta}_l] = \arg\min_{\tau_l, \beta_l} F_l, \tag{42}$$

where

$$F_l \triangleq ||\check{\mathbf{d}}_1 - \beta_l \check{\mathbf{a}}_1(\tau_l)||^2 + ||\check{\mathbf{d}}_2 - \beta_l \check{\mathbf{a}}_2(\tau_l)||^2. \tag{43}$$

In our estimation problem, β_l, $l = 1, 2, \ldots, L$, is a nuisance parameter which needs to be concentrated out to simplify the optimization problem. By letting $\check{\mathbf{d}} = [\check{\mathbf{d}}_1^T \;\; \check{\mathbf{d}}_2^T]^T$ and $\check{\mathbf{a}}(\tau_l) = [\check{\mathbf{a}}_1^T(\tau_l) \;\; \check{\mathbf{a}}_2^T(\tau_l)]^T$, we have

$$\begin{aligned} F_l &= ||\check{\mathbf{d}} - \beta_l \check{\mathbf{a}}(\tau_l)||^2 \\ &= ||\check{\mathbf{d}}||^2 - \beta_l \check{\mathbf{d}}^H \check{\mathbf{a}}(\tau_l) - \beta_l^* \check{\mathbf{a}}^H(\tau_l) \check{\mathbf{d}} + |\beta_l|^2 ||\check{\mathbf{a}}(\tau_l)||^2 \\ &= ||\check{\mathbf{a}}(\tau_l)||^2 \left| \beta_l - \frac{\check{\mathbf{a}}^H(\tau_l) \check{\mathbf{d}}}{||\check{\mathbf{a}}(\tau_l)||^2} \right|^2 + ||\check{\mathbf{d}}||^2 - \frac{|\check{\mathbf{a}}^H(\tau_l) \check{\mathbf{b}}|^2}{||\check{\mathbf{a}}(\tau_l)||^2}. \end{aligned} \tag{44}$$

Let $\beta_l = \check{\mathbf{a}}^H(\tau_l) \check{\mathbf{d}} / \|\check{\mathbf{a}}(\tau_l)\|^2$ and

$$J(\tau_l) = \frac{|\check{\mathbf{a}}^H(\tau_l) \check{\mathbf{b}}|^2}{||\check{\mathbf{a}}(\tau_l)||^2}. \tag{45}$$

Then, the estimate of τ_l, $l = 1, 2, \ldots, L$, can be determined by

$$\hat{\tau}_l = \arg\max_{\tau_l} J(\tau_l). \tag{46}$$

We remark that, due to (46), we can perform the same one-dimensional search for each τ_l, $l = 1, 2, \ldots, L$. Combining the L one-dimensional searches into one, we have the new MLSML estimator, which obtains the estimate of τ_l, $l = 1, 2, \ldots, L$, by finding the arguments corresponding to the L largest local maximums of $J(\tau)$.

The aforementioned one-dimensional search over the parameter space for the L largest peaks is still computationally heavy. To avoid this burdensome task, we use the approach of [7] to obtain a closed-form solution for each $p \in \{0, 1, \ldots, N-1\}$, as detailed in the following.

From the definitions of $\mathbf{a}_1(\tau_l)$ and $\mathbf{a}_2(\tau_l)$, we have

$$\begin{aligned}\check{\mathbf{a}}_1(\tau_l) &= \hat{\mathbf{Q}}^{-1/2}\left[\mathbf{a}_1(pT_C)\ \ \mathbf{a}_1((p+1)T_C)\right]\begin{bmatrix}\nu\\ \mu\end{bmatrix} = \check{\mathbf{A}}_1(pT_C)\boldsymbol{\mu},\\ \check{\mathbf{a}}_2(\tau_l) &= \hat{\mathbf{Q}}^{-1/2}\left[\mathbf{a}_2(pT_C)\ \ \mathbf{a}_2((p+1)T_C)\right]\begin{bmatrix}\nu\\ \mu\end{bmatrix} = \check{\mathbf{A}}_2(pT_C)\boldsymbol{\mu},\end{aligned} \tag{47}$$

where $\boldsymbol{\mu} \triangleq [\nu\ \ \mu]^T = [1-\mu\ \ \mu]^T$,

$$\begin{aligned}\check{\mathbf{A}}_1(pT_C) &\triangleq \hat{\mathbf{Q}}^{-1/2}\left[\mathbf{a}_1(pT_C)\ \ \mathbf{a}_1((p+1)T_C)\right],\\ \check{\mathbf{A}}_2(pT_C) &\triangleq \hat{\mathbf{Q}}^{-1/2}\left[\mathbf{a}_2(pT_C)\ \ \mathbf{a}_2((p+1)T_C)\right].\end{aligned} \tag{48}$$

For notational simplicity, we will drop pT_C from $\check{\mathbf{A}}_1$ and $\check{\mathbf{A}}_2$ as well as $\{k,l\}$ from τ. Plugging (47) into (46), we have

$$\begin{aligned}J(\tau) &= \frac{\boldsymbol{\mu}^T\left(\check{\mathbf{A}}_1^H\check{\mathbf{d}}_1 + \check{\mathbf{A}}_2^H\check{\mathbf{d}}_2\right)\left(\check{\mathbf{A}}_1^H\check{\mathbf{d}}_1 + \check{\mathbf{A}}_2^H\check{\mathbf{d}}_2\right)^H\boldsymbol{\mu}}{\boldsymbol{\mu}^T\left(\check{\mathbf{A}}_1^H\check{\mathbf{A}}_1 + \check{\mathbf{A}}_2^H\check{\mathbf{A}}_2\right)\boldsymbol{\mu}}\\ &= \frac{\boldsymbol{\mu}^T\mathbf{G}\boldsymbol{\mu}}{\boldsymbol{\mu}^T\mathbf{H}\boldsymbol{\mu}} \triangleq \frac{N(\mu)}{D(\mu)},\end{aligned} \tag{49}$$

where

$$\begin{aligned}\mathbf{G} &\triangleq \left(\check{\mathbf{A}}_1^H\check{\mathbf{d}}_1 + \check{\mathbf{A}}_2^H\check{\mathbf{d}}_2\right)\left(\check{\mathbf{A}}_1^H\check{\mathbf{d}}_1 + \check{\mathbf{A}}_2^H\check{\mathbf{d}}_2\right)^H,\\ \mathbf{H} &\triangleq \check{\mathbf{A}}_1^H\check{\mathbf{A}}_1 + \check{\mathbf{A}}_2^H\check{\mathbf{A}}_2.\end{aligned} \tag{50}$$

We know from (49) that $J(\tau)$ is a rational function of two second-order polynomials. Any extreme point $\hat{\mu}$ in the differentiable region of $J(\tau)$ must satisfy the equation

$$S(\mu) = N'(\mu)D(\mu) - N(\mu)D'(\mu) = 0. \tag{51}$$

Note that both $\mathbf{G}$ and $\mathbf{H}$ are Hermitian. Letting g_{ij} and h_{ij} be the ijth elements of $\mathbf{G}$ and $\mathbf{H}$, respectively, we have

$$\begin{aligned}N(\mu) &= (g_{11}+g_{22}-2\bar{g}_{12})\mu^2 + 2(\bar{g}_{12}-g_{11})\mu + g_{11}\\ &\triangleq n_2\mu^2 + n_1\mu + n_0,\\ D(\mu) &= (h_{11}+h_{22}-2\bar{h}_{12})\mu^2 + 2(\bar{h}_{12}-h_{11})\mu + h_{11}\\ &\triangleq d_2\mu^2 + d_1\mu + d_0.\end{aligned} \tag{52}$$

Plugging

$$\begin{aligned}N'(\mu) &= 2n_2\mu + n_1,\\ D'(\mu) &= 2d_2\mu + d_1\end{aligned} \tag{53}$$

into (51) yields

$$\begin{aligned}S(\mu) &= (n_2d_1 - n_1d_2)\mu^2 + 2(n_2d_0 - n_0d_2)\mu + n_1d_0 - n_0d_1\\ &\triangleq s_2\mu^2 + s_1\mu + s_0,\end{aligned} \tag{54}$$

which has the following roots:

$$\hat{\mu} = \frac{-s_1 \pm \sqrt{s_1^2 - 4s_0s_2}}{2s_2}. \tag{55}$$

The MLSML algorithm for code-timing estimation for the user of interest, the kth user, in multipath channels with the time delays within $[0,T)$ can be summarized as follows.

Step 1. Compute $\hat{\mathbf{D}}$ and $\hat{\mathbf{Q}}$ using (29) and (31), respectively.

Step 2. Calculate $\hat{\mu}$ using (55) for each $p \in \{0,1,\ldots,N-1\}$.

Step 3. Calculate the cost function $J(\tau)$ in (49) for the roots obtained in the previous step. In addition, calculate the cost function $J(\tau)$ for each in-differentiable point, that is, $\tau = 0, T_C, \ldots, (N-1)T_C$.

Step 4. Find the L arguments corresponding to the L largest peaks as the estimates of the L resolvable delays.

Now, we relax the constraint that all the time delays are within $[0,T)$. Among the L resolvable delays, we assume $L^{(0)}$ of them are within $[0,T)$ and the others are within $[T,2T)$. For the former $L^{(0)}$ paths, the model derived in the previous section remains the same. As for the later paths, from (8), we know that if $N \le p_l \le 2N-1$, we have, similarly to (12),

$$\begin{aligned}\mathbf{r}_l(m+1) = &\left[\nu\mathbf{J}_1(p)\mathbf{c} + \mu\mathbf{J}_1(p+1)\mathbf{c}\ \ \nu\mathbf{J}_2(p)\mathbf{c} + \mu\mathbf{J}_2(p+1)\mathbf{c}\right]\\ &\times\begin{bmatrix}d(m-1)\\ d(m)\end{bmatrix},\end{aligned} \tag{56}$$

where $p = p_l - N$ and (17) becomes

$$\mathbf{r}_l(m+1) = \left[\mathbf{a}_1(\tau_l - T)\ \ \mathbf{a}_2(\tau_l - T)\right]\mathbf{z}(m). \tag{57}$$

By letting

$$\mathbf{D}^{(0)} = \left[\mathbf{A}_1^{(0)}\boldsymbol{\beta}^{(0)}\ \ \mathbf{A}_2^{(0)}\boldsymbol{\beta}^{(0)}\right], \qquad \mathbf{D}^{(1)} = \left[\mathbf{A}_1^{(1)}\boldsymbol{\beta}^{(1)}\ \ \mathbf{A}_2^{(1)}\boldsymbol{\beta}^{(1)}\right], \tag{58}$$

where

$$\begin{aligned}\mathbf{A}_1^{(0)} &= \left[\mathbf{a}_1(\tau_1)\ \ \mathbf{a}_1(\tau_2)\ \ \cdots\ \ \mathbf{a}_1(\tau_{L^{(0)}})\right],\\ \mathbf{A}_2^{(0)} &= \left[\mathbf{a}_2(\tau_1)\ \ \mathbf{a}_2(\tau_2)\ \ \cdots\ \ \mathbf{a}_2(\tau_{L^{(0)}})\right],\\ \mathbf{A}_1^{(1)} &= \left[\mathbf{a}_1(\tau_{L^{(0)}+1} - T)\ \ \cdots\ \ \mathbf{a}_1(\tau_L - T)\right],\\ \mathbf{A}_2^{(1)} &= \left[\mathbf{a}_2(\tau_{L^{(0)}+1} - T)\ \ \cdots\ \ \mathbf{a}_2(\tau_L - T)\right],\\ \boldsymbol{\beta}^{(0)} &= \left[\beta_1\ \ \beta_2\ \ \cdots\ \ \beta_{L^{(0)}}\right]^T,\\ \boldsymbol{\beta}^{(1)} &= \left[\beta_{L^{(0)}+1}\ \ \beta_{L^{(0)}+2}\ \ \cdots\ \ \beta_L\right]^T,\end{aligned} \tag{59}$$

the data model of (23) becomes

$$\mathbf{r}(m) = \mathbf{D}^{(0)}\mathbf{z}(m) + \mathbf{D}^{(1)}\mathbf{z}(m-1) + \mathbf{w}(m), \quad m = 2,3,\ldots,M. \tag{60}$$

It seems that estimating code timing based on (60) is much more complicated than that based on (23) since $\mathbf{z}(m)$ and $\mathbf{z}(m-1)$ are not independent with each other (cf. (22)). This can be true if we were using an exact ML approach. For MLSML, an approximate ML approach (cf. (38)), the computations are only reasonably increased—Steps 1, 2, and 3 in the MLSML algorithm are performed twice before going to Step 4. For the first round calculation, we calculate based on

$\hat{\mathbf{D}}^{(0)}$ and $\hat{\mathbf{Q}}^{(0)}$ ($\hat{\mathbf{Q}}^{(0)}$ corresponds to $\hat{\mathbf{D}}^{(1)}\mathbf{z}(m-1)$ plus $\mathbf{w}(m)$) with respect to $\mathbf{z}(m)$, and for the second round, we calculate the code-timing based on $\hat{\mathbf{D}}^{(1)}$ and $\hat{\mathbf{Q}}^{(1)}$ ($\hat{\mathbf{Q}}^{(1)}$ corresponds to $\hat{\mathbf{D}}^{(0)}\mathbf{z}(m)$ plus $\mathbf{w}(m)$) with respect to $\mathbf{z}(m-1)$.

We remark that for a given resolvable multipath channel, the performance for the $[0, 2T)$ case can be better than that for the $[0, T)$ case (these two cases are due to different initial conditions). The reason is as follows. For the former case, the cross-correlation between the first $L^{(0)}$ paths in $[0, T)$ and the remaining $L - L^{(0)}$ paths in $[T, 2T)$ can be seen as (cf. (32) and (22))

$$\mathbf{D}^{(0)}\mathrm{E}[\mathbf{z}(m)\mathbf{z}^H(m-1)]\mathbf{D}^{(1)} = \frac{1}{4}\mathbf{D}^{(0)}\begin{bmatrix}1 & 1\\ -1 & -1\end{bmatrix}\mathbf{D}^{(1)}, \quad (61)$$

whereas for the latter case, the cross-correlation between the first $L^{(0)}$ and the remaining $L - L^{(0)}$ paths (all in $[0, T)$) can be seen as (if we still use $\mathbf{D}^{(0)}$ and $\mathbf{D}^{(1)}$)

$$\mathbf{D}^{(0)}\mathrm{E}[\mathbf{z}(m)\mathbf{z}^H(m)]\mathbf{D}^{(1)} = \mathbf{D}^{(0)}\mathbf{D}^{(1)}. \quad (62)$$

The cross-correlation of (61) is lower than that of (62) since the correlation between $\mathbf{z}(m)$ and $\mathbf{z}(m-1)$ is lower than the correlation of $\mathbf{z}(m)$ itself. This performance difference can be seen clearly in the simulation examples given in the next section.

4. SIMULATION RESULTS

In this section, we provide numerical examples to show the performance of the proposed MLSML code-timing estimator over the resolvable multipath channels. Note that MLSML is an extension of LSML which is, in the flat fading case, superior to the other estimators such as the correlator [4], the minimum mean-square-error-based code-timing estimator [11], and the MUSIC code-timing estimator [5, 6] (see [7]). Here, we only provide simulation examples showing the performance of MLSML in the resolvable multipath channel case, along with a comparison with the correlator, which can also be used in the resolvable multipath channel case, whenever applicable. (We do not compare with the method in [8] since (a) the latter can work only for the $[0, T)$ case and (b) the latter has orders of magnitude slower than the former.)

The simulation conditions are as follows. Each user is assigned a Gold sequence of $N = 31$. All users have two paths, that is, $L_k = L = 2$, $k = 1, 2, \ldots, K$. For the user of interest, say, user one, the time delays are fixed to $\tau_{1,1} = 20.5\,T_C$ and $\tau_{1,2} = 23.5\,T_C$, respectively, in the $[0, T)$ case (referred to as Case 1 in the sequel), and $\tau_{1,1} = 29.5\,T_C$ and $\tau_{1,2} = 32.5\,T_C$, respectively, in the $[0, 2T)$ case (referred to as Case 2 in the sequel), with the received powers being $|\beta_{1,1}|^2 = -3$ dB and $|\beta_{1,2}|^2 = 0$ dB, respectively. The other users have random received powers with log-normal distribution. The average received power of each path of the interfering signals is d dB (to be specified in various examples, refereed to as the near-far ratio (NFR)) above the desired user with a standard deviation being 10 dB, that is, $|\beta_{k,l}|^2 = 10^{\rho_{k,l}/10}$, where $\rho_{k,l} \sim N(d, 100)$, $k = 2, 3, \ldots, K$, $l = 1, 2$. The additive noise for $r(i)$ in (7) is white circularly symmetric complex Gaussian with zero-mean and variance $\sigma_n^2 = NN_0/E_s$, where E_s/N_0 is the signal-to-noise ratio (SNR) with E_s being the energy per symbol for user one. (In the simulations, we assume $E_s = N|\beta_{1,2}|^2$.)

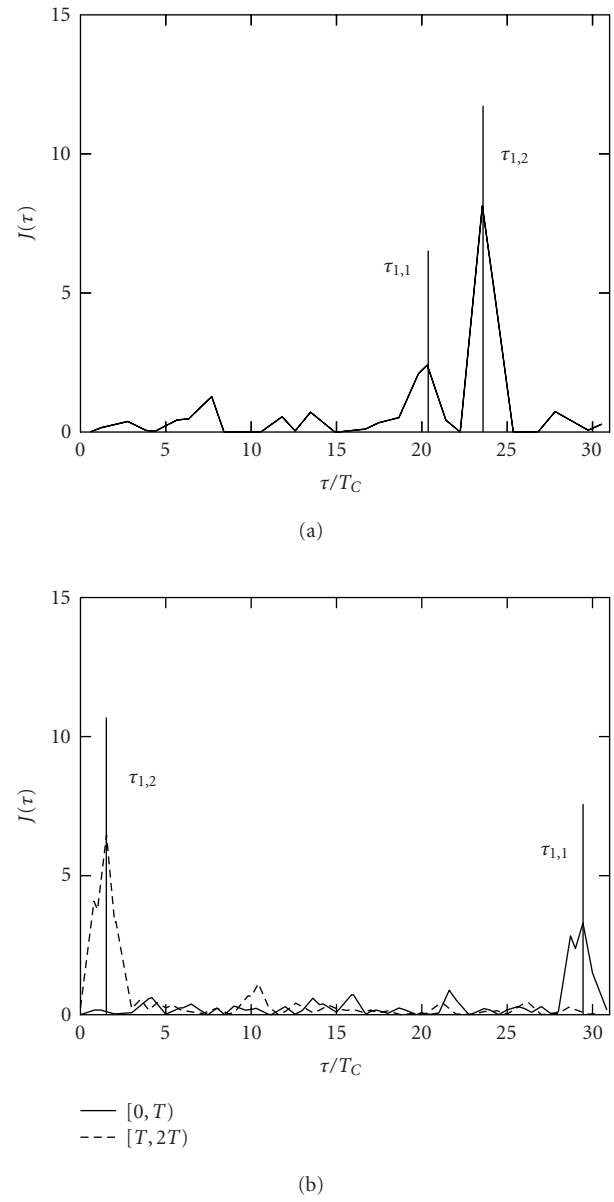

FIGURE 1: Illustration of $J(\tau)$ for (a) $\tau_{1,1}, \tau_{1,2} \in [0, T)$ (Case 1) and (b) $\tau_{1,1} \in [0, T)$, $\tau_{1,2} \in [T, 2T)$ (Case 2) when $K = 10$, $d = 2$, SNR = 10 dB, and $M = 100$.

Before providing the simulation examples to show the performance of the new MLSML code-timing estimator, we look at two figures illustrating $J(\tau)$ of (49) in Case 1 and Case 2, respectively. The specific simulation parameters are as follows: $K = 10$, $d = 2$, SNR = 10 dB, and $M = 100$. The results are shown in Figures 1a, and 1b, respectively.

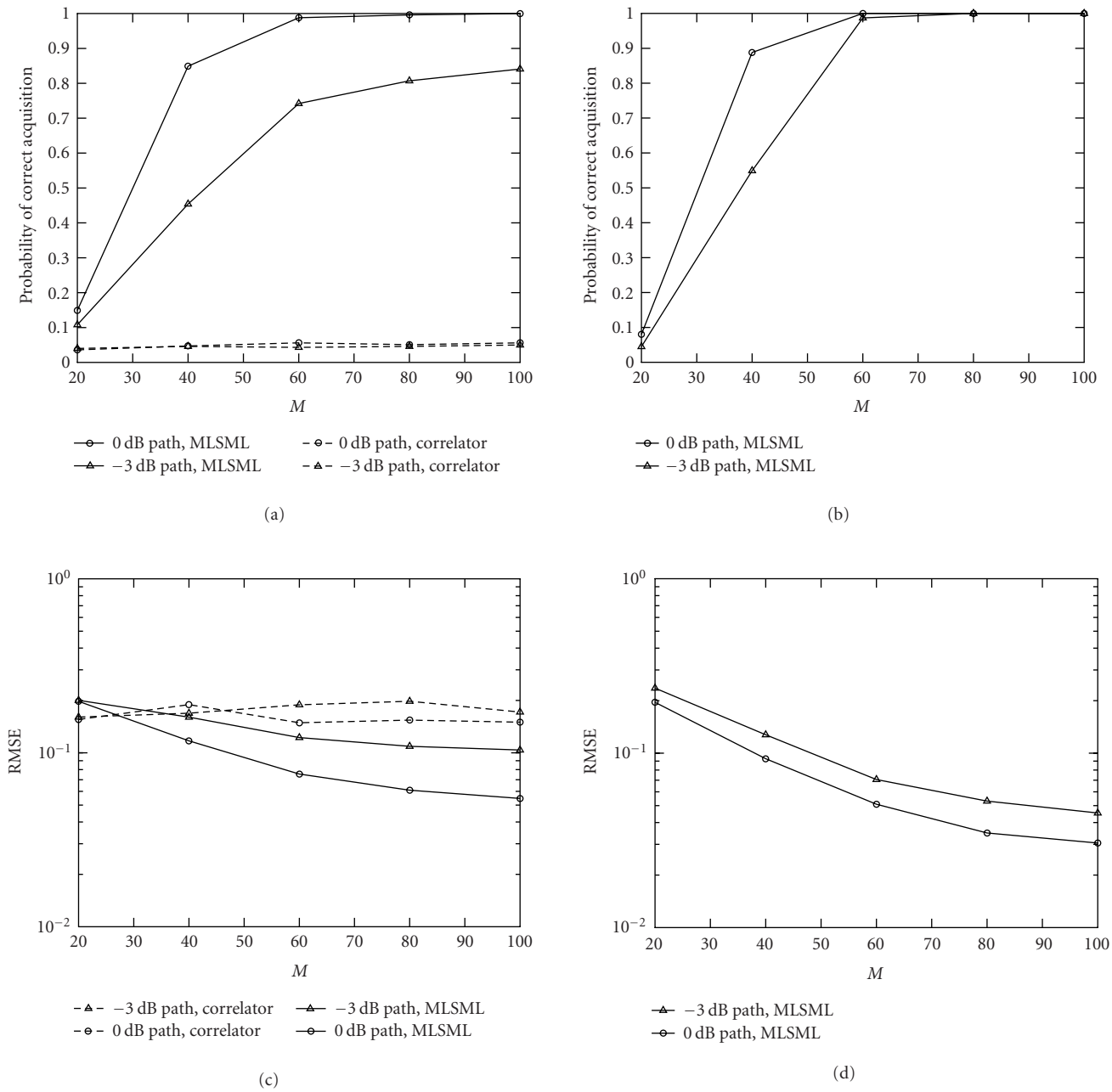

FIGURE 2: Probability of correct acquisition and RMSE versus M for (a) and (c) Case 1, and (b) and (d) Case 2 when $K = 10$, $d = 20$, and SNR = 10 dB.

We can see that we have the correct acquisition in both cases. (A code-timing estimate $\hat{\tau}_{k,l}$ provides a correct acquisition if $|\hat{\tau}_{k,l} - \tau_{k,l}| < T_C/2$.)

Now, we provide simulation examples to show the performance of the proposed code-timing estimator in two aspects: the probability of correct acquisition and the root mean-squared error (RMSE) (normalized by T_C) of the code-timing estimate given correct acquisition. The performance results shown below are based on 1000 Monte Carlo trials. The data symbols and the parameters $\{\tau_{k,l}, \beta_{k,l}\}$ for all the MAI users change from Monte Carlo to Monte Carlo.

Example 1 (performance versus M). The specific simulation parameters are as follows: $K = 10$, $d = 20$, SNR = 10 dB, and the number of data bits M changes from 20 to 100. Figures 2a and 2b show the probability of correct acquisition for Case 1 and Case 2, respectively, while Figures 2c and 2d show their RMSE counterparts. We can see from the figures that the performance of Case 2 is better than that of Case 1

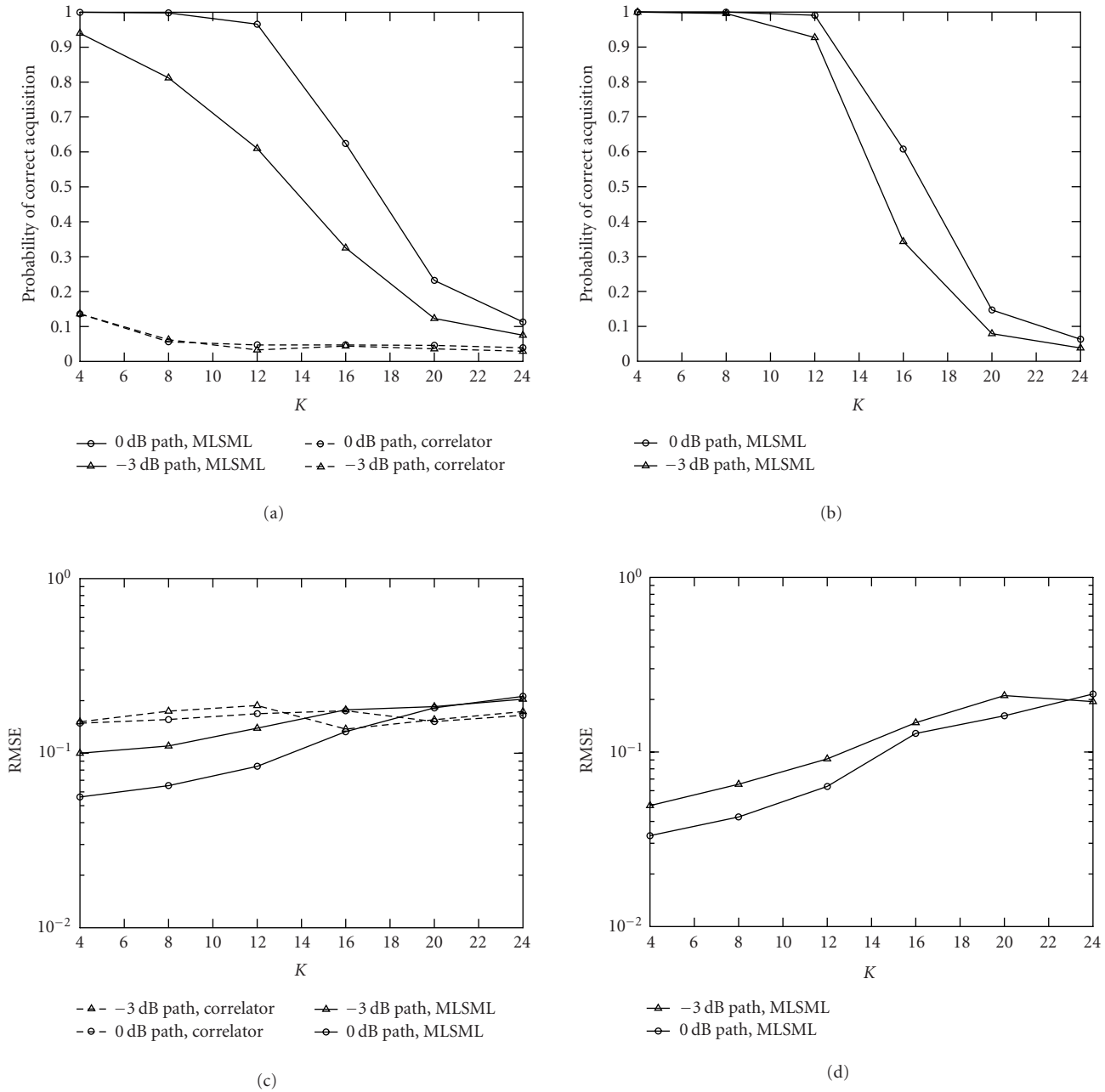

FIGURE 3: Probability of correct acquisition and RMSE versus K for (a) and (c) Case 1, and (b) and (d) Case 2 when $M = 60$, $d = 20$, and SNR = 10 dB.

in that the two paths, especially the weak signal path, have better performance. We can also see that $M = 60 \simeq 2N$ is a threshold for the performance, which conforms with the analysis of [12] showing that when the sample number is twice the number of sensors, the performance of the array using the estimated covariance matrix is within 3 dB of the performance of the array using the exact covariance matrix. (The data model employed in this paper is equivalent to an array with N sensors.) We also provide corresponding curves of the correlator for Case 1 (it does not work for Case 2). We observe that MLSML significantly outperforms the correlator.

Example 2 (performance versus K). The parameters are similar to those in the previous example with the exception that $M = 60$ and K changes from 4 to 24. The results are shown in Figures 3a, 3b, 3c, and 3d, respectively. We can see from the figures that, similar to the previous example,

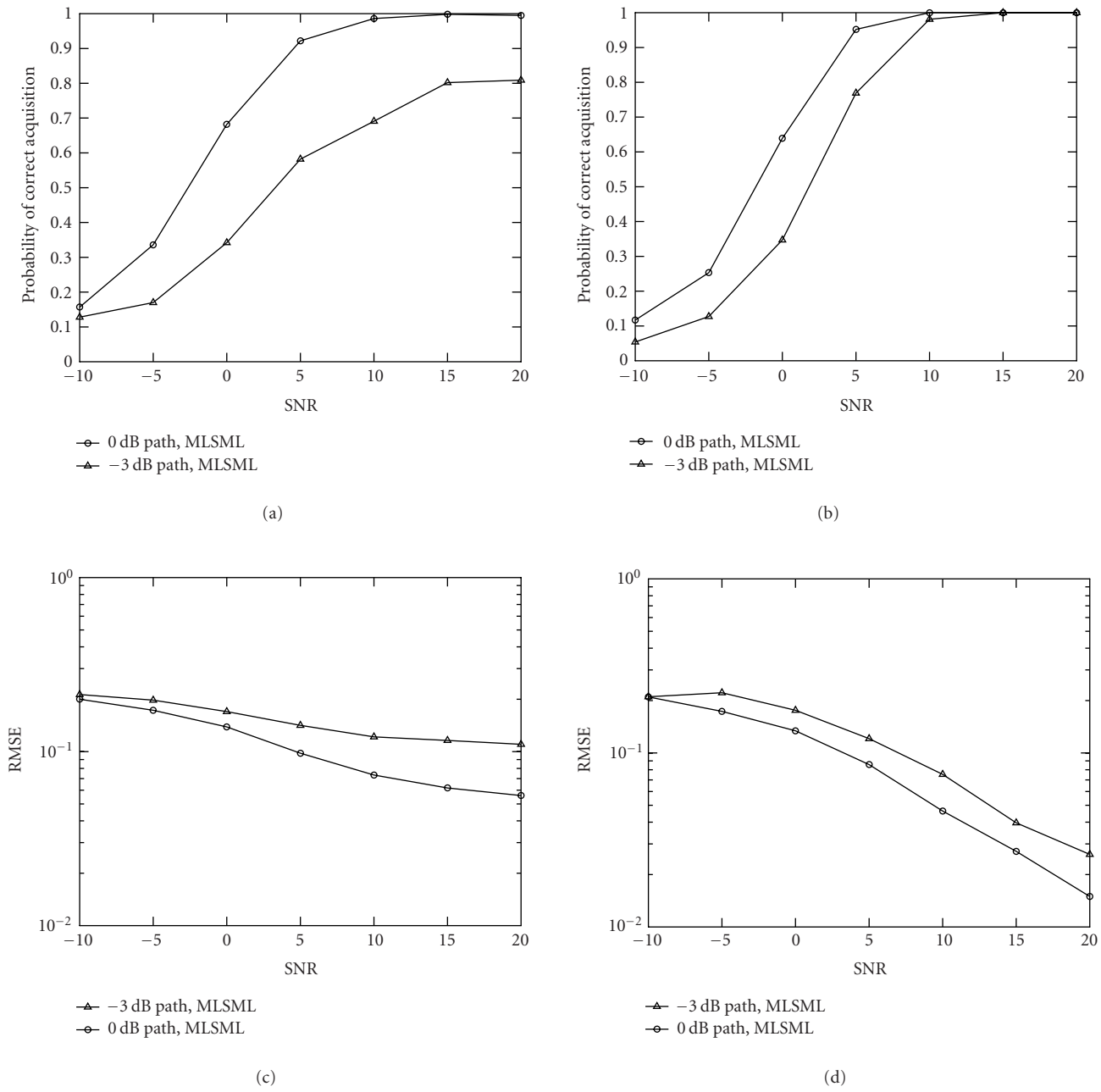

FIGURE 4: Probability of correct acquisition and RMSE versus SNR for (a) and (c) Case 1, and (b) and (d) Case 2 when $M = 60$, $K = 10$, and $d = 20$.

the performance of Case 2 is better than that of Case 1. Note that when K changes from 12 to 16, there is significant performance degradation. The reason is that when $K = 16$, the overall signal number, $LK = 32$, is greater than $N - 1$, the degrees of freedom. (Still, the data model employed herein is equivalent to an array with N sensors which has the degrees of freedom $N - 1$.) We also provide the corresponding curves of the correlator for Case 1. Again, we observe that MLSML significantly outperforms the correlator. Due to the huge performance gap between MLSML and the correlator, we do not need to compare them any further.

Example 3 (performance versus SNR). The parameters are similar to those in the previous example with the exception that $K = 10$ and SNR varies from -10 dB to 20 dB. The performance of the MLSML estimator is shown in Figures 4a, 4b, 4c, and 4d, respectively. Still, the performance for Case 2 is better than that for Case 1.

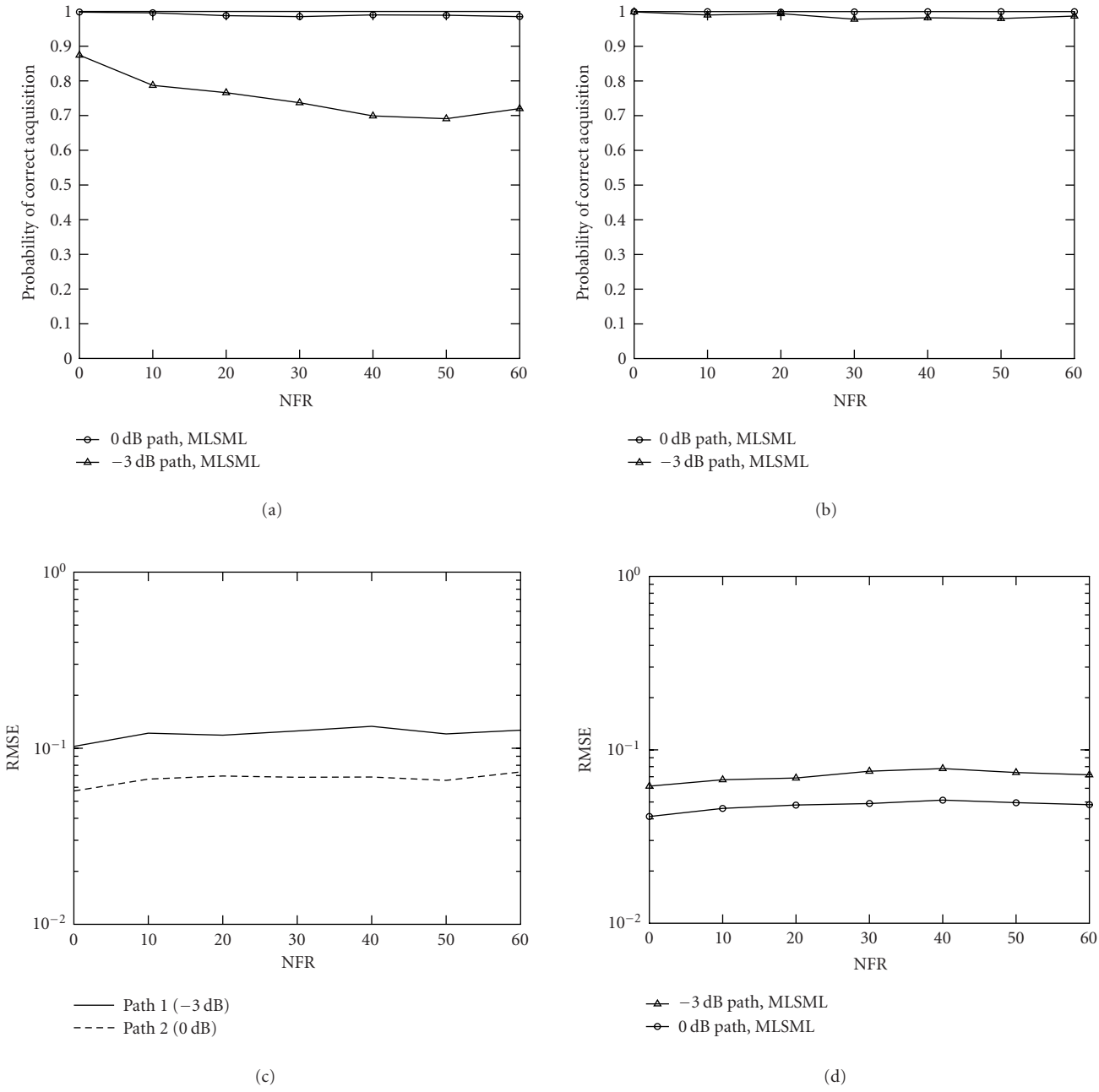

Figure 5: Probability of correct acquisition and RMSE versus d for (a) and (c) Case 1, and (b) and (d) Case 2 when $M = 60$, $K = 10$, and SNR = 10 dB.

Example 4 (performance versus d). This example shows the near-far resistant capability of the new code-timing estimator. Figures 5a, 5b, 5c, and 5d, and Figures 6a, 6b, 6c, and 6d show the performance with $K = 10$ and $K = 15$, respectively. The other parameters are similar to those in the previous example with the exception that d varies from 0 to 60. We can see from the figures that, for $K = 10$, where the interference number $L(K - 1) = 18$ is much less than N, we experience almost no performance degradation when d increases (only a minor degradation for the weak path in Case 1). However, for $K = 15$, where the interference number $L(K - 1) = 28$ is only slightly less than $N - 1$, we experience significant performance degradation when d increases. These simulations suggest that with the reasonable selection of parameters, such as $\sum_{k=1}^{K} L_k < N$ (which can be seen as a fifth assumption in addition to those given at the end of Section 2), MLSML can have a quite high probability of correct acquisition even when the near-far problem is severe.

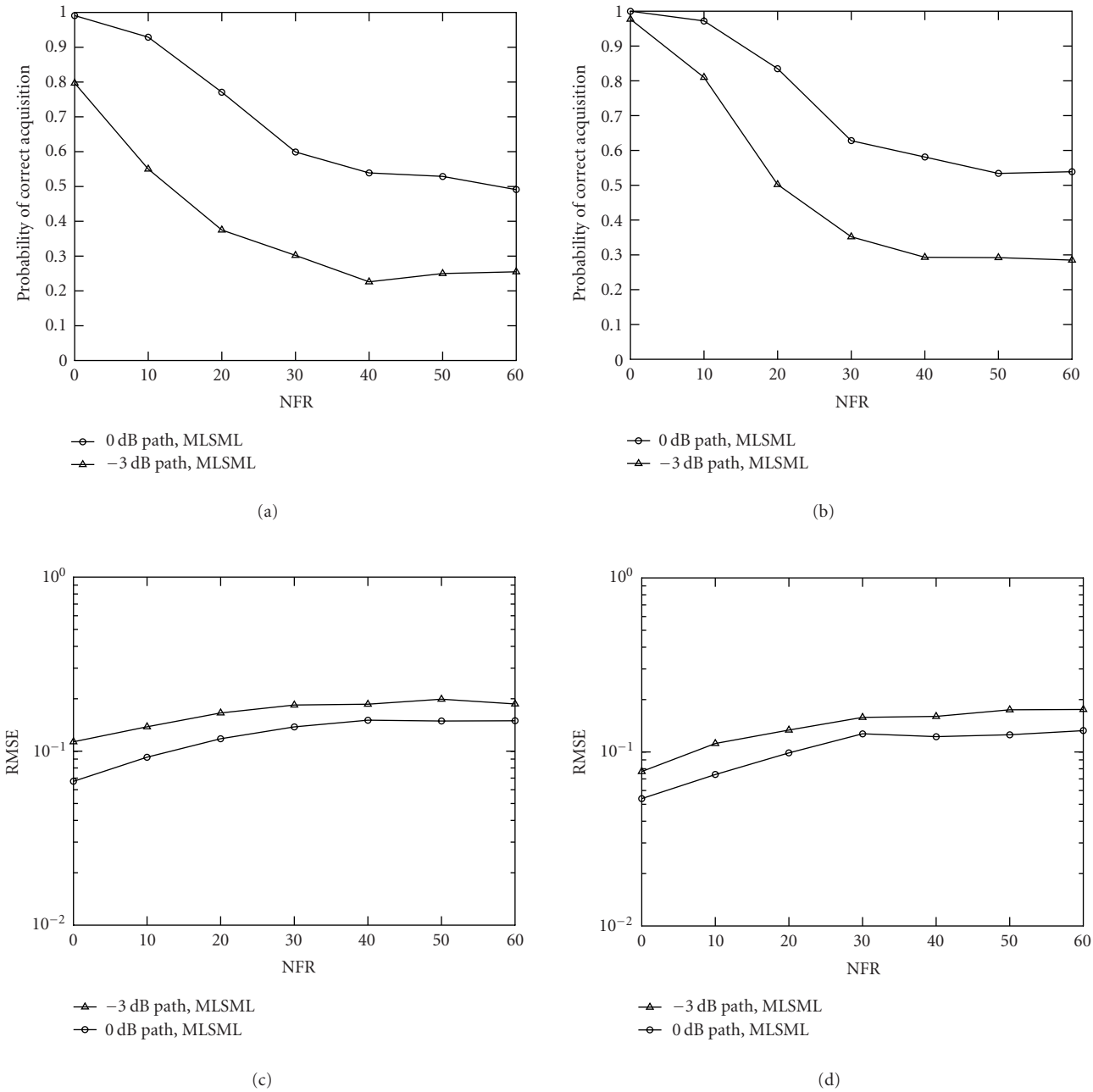

FIGURE 6: Probability of correct acquisition and RMSE versus d for (a) and (c) Case 1, and (b) and (d) Case 2 when $M = 60$, $K = 15$, and SNR = 10 dB.

5. CONCLUDING REMARKS

We have presented an MLSML estimator that can be used to perform the code-timing estimation for the DS-CDMA systems over the resolvable multipath channels in a closed form. Simulation examples have shown that MLSML can be used to provide a high correct acquisition probability and a high estimation accuracy. Simulation examples have also shown that MLSML can have very good near-far resistant capability, due to employing a data model similar to that for adaptive array processing where strong interferences can be suppressed within the capability of the degrees of freedom of the array.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation Grant CCR-0097114.

REFERENCES

[1] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA, 3rd edition, 1995.

[2] J. C. Liberti Jr. and T. S. Rappaport, *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*, Prentice-Hall, Upper Saddle River, NJ, USA, 1999.

[3] 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., pp. 59–70, Springer-Verlag, New York, NY, USA, 1989.

[4] R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to Spread Spectrum Communications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.

[5] E. G. Strom, 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, no. 1, pp. 84–93, 1996.

[6] S. E. Bensley and B. Aazhang, "Subspace-based channel estimation for code division multiple access communication systems," *IEEE Trans. Commun.*, vol. 44, no. 8, pp. 1009–1020, 1996.

[7] D. Zheng, J. Li, S. L. Miller, and E. G. storm, "An efficient code-timing estimator for DS-CDMA signals," *IEEE Trans. Signal Processing*, vol. 45, no. 1, pp. 82–89, 1997.

[8] G. Ye and G. Bi, "Code timing estimator for DS-CDMA signals in slow fading multipath channels," *Electronics Letters*, vol. 35, no. 19, pp. 1604–1606, 1999.

[9] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1997.

[10] J. Li, B. Halder, P. Stoica, and M. Viberg, "Computationally efficient angle estimation for signals with known waveforms," *IEEE Trans. Signal Processing*, vol. 43, no. 9, pp. 2154–2163, 1995.

[11] R. F. Smith and S. L. Miller, "Code timing estimation in a near-far environment for direct-sequence code-division multiple-access," in *Proc. IEEE Military Communications Conference (MILCOM '94)*, pp. 47–51, Fort Monmouth, NJ , USA, October 1994.

[12] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 10, no. 6, pp. 853–863, 1974.

Jianhua Liu received the B.S. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1984, the M.S. degree in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 1987, the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 1998, and the Ph.D. degree majoring in electrical engineering and minoring in statistics from University of Florida in 2004. From March 1987 to February 1999, he worked at the Communication, Telemetry and Telecontrol Research Institute, Shijiazhuang, China, where he was an Assistant Engineer, Engineer, Senior Engineer, and Fellow Engineer. From March 1995 to August 1998, he was also a Research Assistant at Tsinghua University. From February 1999 to June 2000, he worked at Nanyang Technological University, Singapore, as a Research Fellow. From July 2000 to August 2004, he was a Research Assistant at the University of Florida. Since August 2004, he has been an Assistant Professor of electrical engineering at Embry-Riddle Aeronautical University, Daytona Beach campus. His research interests include wireless communications, signal processing, and avionics.

Jian Li received the M.S. 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 in the Department of Electrical Engineering, The Ohio State University, Columbus. From July 1991 to June 1993, she was an Assistant Professor in the Department of Electrical Engineering, University of Kentucky, Lexington. Since August 1993, she has been with the Department of 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 their applications. Dr. Li is a Member of Sigma Xi and Phi Kappa Phi. She received the 1994 National Science Foundation Young Investigator Award and the 1996 Office of Naval Research Young Investigator Award. She was an Executive Committee Member of the 2002 International Conference on Acoustics, Speech, and Signal Processing, Orlando, Florida, May 2002. She has been an Associate Editor of the IEEE Transactions on Signal Processing since 1999 and an Associate Editor of the IEEE Signal Processing Magazine since 2003. She is presently a Member of the Signal Processing Theory and Methods (SPTM) Technical Committee of the IEEE Signal Processing Society.