Blind Decorrelating RAKE Receivers for Long-Code WCDMA

Lang Tong, *Senior Member, IEEE*, Alle-Jan van der Veen, *Senior Member, IEEE*, Patrick Dewilde, *Fellow, IEEE*, and Youngchul Sung, *Student Member, IEEE*

***Abstract*—The problem of blind and semiblind channel estimation and symbol detection is considered for long-code wideband code division multiple access (CDMA) systems, including systems with multirate and multicode transmissions. A decorrelating matched filter, implemented efficiently in state-space, eliminates multiaccess interference and produces a bank of vector processes. Each vector process spans a one-dimensional (1-D) subspace from which channel parameters and data symbols of one user are estimated jointly by least squares. A new identifiability condition is established, which suggests that channels unidentifiable in short-code CDMA systems are almost surely identifiable when aperiodic spreading codes are used. The decorrelating matched filter is implemented efficiently based on time-varying state-space realizations that exploit the structure of sparsity of the code matrix. The mean square error of the estimated channel is compared to the Cramér–Rao bound, and a bit error rate (BER) expression for the proposed algorithm is presented.**

***Index Terms*—Aperiodic spreading sequences/codes, blind and semiblind multiuser detection, channel estimation, decorrelating matched filter/RAKE receiver, fast algorithms, identifiability conditions, long-code wideband CDMA.**

I. Introduction

A. Problem and the Approach

WE CONSIDER the problem of joint channel estimation and symbol detection in a long-code wideband code division multiple access (CDMA) system that has features of third-generation wireless. The scrambling sequences are aperiodic, data and control information may be modulated separately onto the in-phase and quadrature parts of the signal using different channelization codes with different spreading gains, pilots are often part of the control symbols, users may have different spreading gains, or multiple channelization codes may be assigned to the same user. For uplink applications, users are asynchronous, and their multipath channels may have delays longer than the symbol period. Multiple antennas may be used.

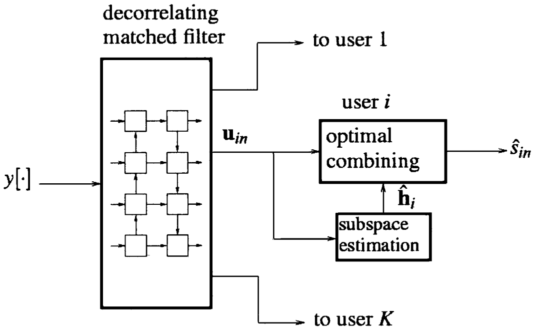


Fig. 1. Receiver structure. The decorrelating matched filter is implemented by an efficient state-space realization.

RAKE receivers are widely used in both uplink and downlink CDMA systems. If the spreading codes have good cross- and auto-correlation properties, the matched filter front-end suppresses multiaccess interference, and the RAKE receiver captures multipath diversity through its diversity branches (or the RAKE fingers). For high-rate CDMA under frequency selective fading, however, code orthogonality can not be guaranteed, and the conventional RAKE receiver that uses a bank of matched filters may perform poorly. The loss of code orthogonality has adverse effects on both channel estimation and symbol detection, and the performance degradation is especially pronounced when the network is heavily loaded and power control imperfect.

In this paper, we propose a joint channel and symbol estimation scheme for RAKE receivers. As illustrated in Fig. 1, a decorrelating matched filter projects the received chip-rate sequence $y[n]$ into the signal space of each user whose channel and data sequence can be estimated jointly and independent of other users by least squares via a rank-one decomposition. The decorrelating matched filter does not depend on channel coefficients and may be precomputed for certain applications. The proposed scheme imposes no conditions on channel parameters and is capable of dealing with rapid multipath fading. We also establish a new identifiability condition that depends only on the spreading codes used in the system but not on channel parameters. Implied by this identifiability condition is that aperiodic spreading codes enhance channel identifiability; channels not identifiable in short-code CDMA are almost surely identifiable in a long-code system.

A key contribution of this work is an efficient implementation of the decorrelating matched filter. The idea of using the

Manuscript received April 17, 2002; revised December 6, 2002. The work of L. Tong was partially carried out as part of a visiting professorship granted by the Cor Wit Foundation. This work was supported in part by the Army Research Office under Grant ARO-DAAB19-00-1-0507 and the Multidisciplinary University Research Initiative (MURI) under the Office of Naval Research Contract N00014-00-1-0564. The work of A. J. van der Veen was supported in part by the Dutch Ministry of Economic Affairs under TSIT 1025 "Beyond-3G." The associate editor coordinating the review of this paper and approving it for publication was Prof. Nicholas D. Sidiropoulos.

L. Tong and Y. Sung are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail ltong@ece.cornell.edu; ys87@ece.cornell.edu).

A.-J. van der Veen and P. Dewilde are with Delft University of Technology, Delft, The Netherlands (e-mail: allejan@cobalt.et.tudelft.nl; dewilde@dimes.tudelft.nl).

Digital Object Identifier 10.1109/TSP.2003.811230

decorrelating matched filter for short-code CDMA is known [1], but applying it to long-code CDMA presents a daunting task in terms of both computational complexity and storage requirements. A direct implementation for a ten user system—each has three multipath fingers with a 100-symbol slot and a spreading gain of 64—amounts to inverting a code matrix of size around 6400×3000. The code matrix, fortunately, is highly structured and sparse; only 1% of its entries are nonzero. The inverse of the code matrix, however, will in general lose the structure and the sparsity. In this paper, we use the extensive theory and algorithms developed by Dewilde and Van der Veen [2], who considered the inversion of infinite size structured matrices. The idea is to replace the code matrix by a time-varying state-space realization and implement the inversion locally in state-space. For cases where precomputation of the code matrix is possible, our approach has the same level of online computations and storage requirements as that of the conventional matched filter. If the inversion of code matrix must be performed online, we are able to reduce the computational complexity to the same level as that required in the short-code case.

B. Related Work

A blind two-dimensional (2-D) RAKE receiver for long-code CDMA was first proposed by Zoltowski *et al.* in [3] and further developed in [4] and [5]. Their approach is perhaps the earliest blind multiuser detector applicable to long-code CDMA. There are similarities between their approach and the one presented in this paper; both use the RAKE concept. The differences, however, are substantial. Zoltowski's two-dimensional (2-D) RAKE uses a conventional matched filter as the first stage, followed by post processing to mitigate multiuser interference. We use a decorrelating matched filter to separate users up front and perform single-user optimal RAKE combining as the second step. The channel is estimated via a matrix pencil technique based on second-order statistics by Zoltowski *et al.* We use deterministic least squares, which has the advantage of requiring a small number of samples. The implementation of our approach is also considerably simpler because of the state-space technique that exploits the special structure of the code matrix.

Blind channel estimation and multiuser detection for long-code CDMA has been considered by a number of other authors. Iterative techniques based on maximizing the likelihood function [6], [7] and least squares [8] have been proposed. These are high-performance techniques but also have well-known drawbacks such as ill convergence and high complexity. They are best complemented by initialization techniques such as the algorithm developed in this paper. In addition, in the literature, we have second-order moment techniques based on the uncorrelated or i.i.d. assumption of the spreading code or the symbols [3], [4], [9]–[11]. These techniques rely on the convergence of time averages to statistical averages, which often requires hundreds to thousands of symbols. The work of Weiss and Friedlander [12] is perhaps the closest to our approach, although they focus on down link applications. They assume that multipath delays are limited to a small fraction of a symbol interval. By dropping samples that contain intersymbol interference, they propose to invert the (reduced) code matrix followed by a different subspace algorithm and an iterative likelihood maximization. Their assumptions imply that their algorithm is not applicable to systems with asynchronous users and long multipath delays. Other related approaches include the use of chip-level equalization. See, for example, [13], [14], and references therein.

C. Notation

We will use the notion of zero-dimensional vectors and matrices. In particular, a matrix or a vector with zero dimension is denoted by "$\bullet$." Normal multiplication and addition rules apply to zero dimensional matrices. Specifically, if $\mathbf{a} = \bullet$ has size $m \times 0$ and $\mathbf{b} = \bullet$ has size $0 \times n$, then $\mathbf{ab}$ is a matrix of size $m \times n$ with all entries equal to 0 (since its rank has to be 0).

Other notations are standard. Vectors and matrices are written in boldface with matrices in capitals. We reserve $\mathbf{I}_m$ for the identity matrix of size m (the subscript is included only when necessary) and $\mathbf{0}_{m\times n}$ for the $m \times n$ zero matrix. For a random vector $\mathbf{x}$, $\mathbb{E}(\mathbf{x})$ is the statistical expectation of $\mathbf{x}$. The notation $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ means that $\mathbf{x}$ is (complex) Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Operations $(\cdot)^T$ and $(\cdot)^H$ indicate transpose and Hermitian transpose, respectively. Given a matrix $\mathbf{X}$, $\mathcal{R}\{\mathbf{X}\}$ is the range space of matrix $\mathbf{X}$, $\mathbf{X}^\dagger$ the Moore–Penrose pseudo inverse, and $\mathbf{X} \otimes \mathbf{Y}$ the Kronecker product of $\mathbf{X}$ and $\mathbf{Y}$. For a matrix (vector) $\mathbf{X}$, we use $\|\mathbf{X}\|_F$ for the Frobenius norm and $\|\mathbf{X}\|$ for the 2-norm.

II. Matrix Model

Matrix models for long-code CDMA have been derived in several papers, e.g., in [12] and [15]; hence, we will make only a brief derivation.[1]

We assume that K asynchronous users transmit linearly modulated symbols. The transmission is slotted, and user i transmits M_i symbols $\{s_{in}, n = 1, \ldots, M_i\}$ in each slot. The symbol sequence from user i is represented by the vector $\mathbf{s}_i \triangleq [s_{i1}, \ldots, s_{i,M_i}]^T$. At the transmitter, each symbol s_{in} is spread by an aperiodic code vector $\mathbf{c}_{in}$ with spreading gain (length) G_i, followed by a chip pulse-shaping filter. The propagation channel of user i can be modeled by an equivalent chip-rate finite impulse response $h_{ij}, j = 0, \ldots, L_i - 1$, where h_{ij} can be viewed as the gain of the jth finger of the user i's multipath channel.

Because the channel is linear, we can first focus on symbol s_{in} from user i transmitted in the nth symbol interval and set all other symbols and noise to zero. Let the received signal corresponding to symbol s_{in} be passed through a chip-matched filter and sampled at the chip rate.[2] All samples are put in a vector $\mathbf{y}_{in}$. As shown in Fig. 2, $\mathbf{y}_{in}$ is a linear combination of shifted (delayed) code vectors $\mathbf{c}_{in}$, where $\mathbf{c}_{in}$ is the segment of G_i chips of user i's spreading code corresponding to the nth symbol. Each shifted code vector is multiplied by the jth fading coefficient h_{ij}, and the channel response to s_{in} is given by

$$\mathbf{y}_{in} = \mathbf{T}_{in}\mathbf{h}_i s_{in}, \qquad \mathbf{h}_i \triangleq [h_{i0}, \ldots, h_{i,L_i-1}]^T.$$

[1] An algebraic derivation of the model based on Nyquist sampling is given in [16].

[2] In general, sufficient statistics can be obtained by Nyquist sampling. The derivation here will correspond to the even (or odd) subsequence of an observation sampled at twice the chip rate.

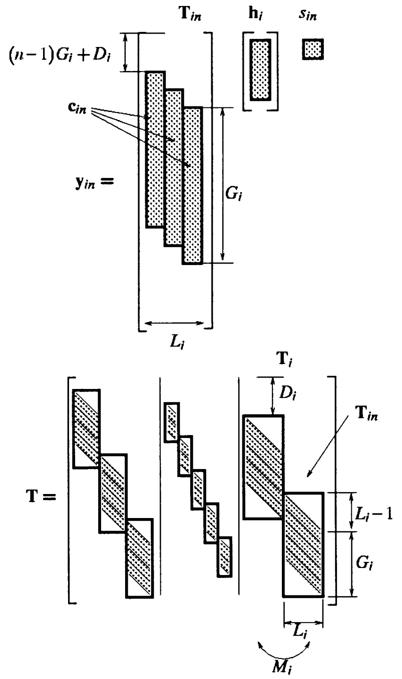


Fig. 2. (Top) Structure of $\mathbf{T}_{in}$. (Bottom) Structure of the code matrix $\mathbf{T}$.

Here, $\mathbf{T}_{in}$ is the code matrix of user i and symbol n [see the top part of Fig. 2], and $\mathbf{h}_i$ is the multipath fading channel for user i. We assume that user i has a relative delay of D_i chips with respect to the reference at the receiver. One can view that each column of $\mathbf{T}_{in}$ corresponds to a discrete multipath component. For example, the first column of $\mathbf{T}_{in}$ is made of $(n-1)G_i+D_i$ zeros that model the relative delay of the first path with respect to the reference followed by the code vector $\mathbf{c}_{in}$ and additional zeros that make the size of $\mathbf{y}_{in}$ the total number of chips of the entire slot. The second column of $\mathbf{T}_{in}$ models the second multipath component similarly. Note that for sparse channels, the shifting of the code vectors does not have to be consecutive.

For user i, the total received noiseless signal is given by

$$\mathbf{y}_i = \sum_{n=1}^{M_i} \mathbf{T}_{in}\mathbf{h}_i s_{in} = \mathbf{T}_i \left(\mathbf{I}_{M_i} \otimes \mathbf{h}_i\right) \mathbf{s}_i$$
$$\mathbf{T}_i \triangleq [\mathbf{T}_{i1}, \ldots, \mathbf{T}_{i,M_i}].$$

Matrix $\mathbf{T}_i$ is the code matrix of user i, and it does not depend on the gains and phases of the multipath channel. Now including all users and the noise, we have

$$\begin{aligned} \mathbf{y} &= \mathbf{THs} + \mathbf{w} \\ \mathbf{T} &\triangleq [\mathbf{T}_1, \ldots, \mathbf{T}_K] \\ \mathbf{H} &\triangleq \text{diag}\left(\mathbf{I}_{M_1} \otimes \mathbf{h}_1, \ldots, \mathbf{I}_{M_K} \otimes \mathbf{h}_K\right) \end{aligned} \tag{1}$$

where matrix $\mathbf{H}$ is block diagonal with $\mathbf{I}_{M_i} \otimes \mathbf{h}_i$ as the ith block, vector $\mathbf{s}$ is a stacking of all symbol vectors, and $\mathbf{w}$ is a vector representing the additive Gaussian noise. The structure of the code matrix $\mathbf{T}$ is illustrated in the bottom part of Fig. 2. Note that by allowing $\mathbf{T}_i$ to have different sizes for different users, we include cases where variable spreading gains are used.

We will impose the following assumptions.

A1) The code matrix $\mathbf{T}$ is known.
A2) The code matrix $\mathbf{T}$ has full column rank.
A3) The noise vector is complex Gaussian $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ with possibly unknown σ^2.

Assumption A1) implies that the receiver knows the codes, the delay offsets D_i, and the number of channel coefficients L_i of all users. If D_i is unknown, we may set it to 0 and model all paths. L_i is a model parameter, and its choice is often left to algorithm designers. Since any channel coefficient is allowed to be zero, one can overparameterize the channel to accommodate channel length and delay uncertainties and pay a price for the lack of modeling details. If we know that the channel is sparse, it is more efficient to model the channel as separate clusters of fingers. In that case, we assume that the approximate locations of these clusters are known.

Assumption A2) is sufficient but not necessary for the channel to be identifiable and for the proposed algorithm to produce good estimates. When A2) fails, the channel may still be identifiable, as shown in Theorem 1 in Section IV. In practice, one may only include a limited number of dominant interferers and significant fingers in the data model. In Section III-D, we consider modifications of the proposed algorithm applicable when the code matrix is singular.

III. Blind and Semiblind Decorrelating Rake

We present in this section a decorrelating RAKE receiver that jointly estimates the channel $\mathbf{h}_i$ and data $\mathbf{s}_i$ of every user i independently. As illustrated in Fig. 1, we use a decorrelating matched filter $\mathbf{T}^\dagger$ as a front-end to remove multiaccess interference. Other types of matched filters can also be used, of course, and are briefly discussed in Section III-D. We will present the details of an efficient time-varying state-space implementation of the decorrelating matched filter in Section V. We note here only that the complexity of the decorrelating RAKE is comparable with that of the conventional one.

A. Blind Channel Estimation via Least Squares

The output of the decorrelating matched filter is given by

$$\mathbf{u} = \mathbf{T}^\dagger\mathbf{y} = \text{diag}(\mathbf{I} \otimes \mathbf{h}_1, \ldots, \mathbf{I} \otimes \mathbf{h}_K)\mathbf{s} + \mathbf{n} \tag{2}$$

where $\mathbf{n} = \mathbf{T}^\dagger\mathbf{w}$ is the (colored) noise vector. Partition $\mathbf{u}$ into segments of length L_i with $\mathbf{u}_{in}$ as the $(\sum_{j=1}^{i-1} M_j) + n$th subvector. The structure of $\mathbf{u}$ in (2) implies that $\mathbf{u}_{in}$ corresponds to symbol n of user i and satisfies

$$\mathbf{u}_{in} = \mathbf{h}_i s_{in} + \mathbf{n}_{in}, \qquad n = 1, \ldots, M_i. \tag{3}$$

Collecting all data for user i gives

$$\mathbf{U}_i = [\mathbf{u}_{i1}, \ldots, \mathbf{u}_{iM_i}] = \mathbf{h}_i\mathbf{s}_i^T + \mathbf{N}_i. \tag{4}$$

Treating $\mathbf{h}_i$ and $\mathbf{s}_i$ as deterministic parameters, we can define the least squares problem

$$\{\mathbf{h}_i, \mathbf{s}_i\} = \arg\min_{\mathbf{h},\mathbf{s}} \|\mathbf{U}_i - \mathbf{hs}^T\|_F^2$$

and estimates of $\mathbf{h}_i$ and $\mathbf{s}_i$ (with an unknown scaling factor) are found from a rank-one approximation of $\mathbf{U}_i$. In other words, denoting

$$\hat{\mathbf{R}}_i \triangleq \frac{1}{M_i}\sum_{n=1}^{M_i} \mathbf{u}_{in}\mathbf{u}_{in}^H \tag{5}$$

we obtain the least squares estimates

$$\hat{\mathbf{h}}_i = \arg\max_{\|\mathbf{g}\|=1} \mathbf{g}^H \hat{\mathbf{R}}_i \mathbf{g}, \qquad \hat{s}_{in} = \hat{\mathbf{h}}_i^H \mathbf{u}_{in}. \tag{6}$$

The solution $\hat{\mathbf{h}}_i$ is given as the dominant eigenvector of $\hat{\mathbf{R}}_i$. The scaling ambiguity in the above estimates must be removed by either incorporating prior knowledge of the symbol, using pilot symbols, or employing differential encoding of s_{in}.

B. Semiblind Channel Estimation

If arbitrarily placed pilot symbols exist in $\mathbf{s}_i$, the above least squares problem can be amended. Let $\mathbf{s}_i$ be partitioned into two subvectors: $\mathbf{s}_{i_p}$ containing the pilot and $\mathbf{s}_{i_d}$ the data. We partition $\mathbf{U}_i$ accordingly in $\mathbf{U}_{i_p}$ and $\mathbf{U}_{i_d}$. The least squares estimator of $\mathbf{h}_i$ is given by

$$\hat{\mathbf{h}}_i = \arg\min_{\mathbf{h},\,\mathbf{s}} \|\mathbf{U}_{i_p} - \mathbf{h}\mathbf{s}_{i_p}^T\|_F^2 + \|\mathbf{U}_{i_d} - \mathbf{h}\mathbf{s}^T\|_F^2. \tag{7}$$

The above optimization does not have a closed-form solution. Simple iterative schemes can be applied. We note that for a fixed $\hat{\mathbf{h}}_i$, the optimal choice of $\mathbf{s}_{i_d}$ is $\hat{\mathbf{s}}_{i_d}^T = \hat{\mathbf{h}}_i^H \mathbf{U}_{i_d}/\|\hat{\mathbf{h}}_i\|^2$. This leads to the following iteration given the estimate $\hat{\mathbf{h}}_i[k]$ at the kth iteration:

$$\begin{aligned}\hat{\mathbf{s}}_{i_d}^T[k] &= \hat{\mathbf{h}}_i^H[k]\mathbf{U}_{i_d}/\|\hat{\mathbf{h}}_i[k]\|^2\\ \hat{\mathbf{h}}_i[k+1] &= \arg\min_{\mathbf{h}} \|\mathbf{U}_{i_p} - \mathbf{h}\mathbf{s}_{i_p}^T\|_F^2 + \|\mathbf{U}_{i_d} - \mathbf{h}\hat{\mathbf{s}}_{i_d}^T[k]\|_F^2\end{aligned}$$

which is equivalent to treating $\hat{\mathbf{s}}_{i_d}[k]$ as known data. One can, of course, make hard decisions on $\hat{\mathbf{s}}_{i_d}[k]$ for further enhancement. Other iterative techniques can also be applied [17].

C. Whitened Estimator

The symbol estimator given in (6) is the standard maximum ratio combining of signals from different RAKE fingers. It is not optimal even if $\hat{\mathbf{h}}_i$ is perfect because it does not take into account that the vector noise process $\mathbf{n}_{in}$ is colored both in n and in its components. If we ignore the coloring in n, then a simple whitening approach can be applied. We know, from (3), that

$$\mathbf{n}_{in} \sim \mathcal{N}(\mathbf{0},\, \sigma^2\boldsymbol{\Sigma}_{in})$$

where $\boldsymbol{\Sigma}_{in}$ is the $L_i \times L_i$ submatrix obtained from the $(\sum_{j=1}^{i-1} M_j) + n$th diagonal block of $\mathbf{T}^\dagger(\mathbf{T}^\dagger)^H$. The whitened RAKE receiver is given by

$$\hat{s}_{in} = \hat{\mathbf{h}}_i^H \boldsymbol{\Sigma}_{in}^{-1}\mathbf{u}_{in}. \tag{8}$$

The channel estimator given in (6) is also affected by the colored noise. However, this coloring is known and can be whitened. Specifically, recall (4) and (5). We have

$$\mathbb{E}(\hat{\mathbf{R}}_i) = \frac{\|\mathbf{s}_i\|^2}{M_i}\mathbf{h}_i\mathbf{h}_i^H + \sigma^2\boldsymbol{\Delta}_i, \qquad \boldsymbol{\Delta}_i \triangleq \frac{1}{M_i}\sum_{n=1}^{M_i}\boldsymbol{\Sigma}_{in}$$

where $\boldsymbol{\Delta}_i$ is a known matrix. The channel can then be estimated from the following modification which whitens the noise on $\hat{\mathbf{R}}_i$:

$$\begin{aligned}\mathbf{g}_* &= \arg\max_{\|\mathbf{g}\|=1} \mathbf{g}^H\left(\boldsymbol{\Delta}_i^{-1/2}\hat{\mathbf{R}}_i\boldsymbol{\Delta}_i^{-H/2}\right)\mathbf{g}\\ \hat{\mathbf{h}}_i &= \boldsymbol{\Delta}^{1/2}\mathbf{g}_*.\end{aligned}$$

D. Other Front-Ends

The decorrelating matched filter $\mathbf{T}^\dagger$ leads to exact channel identification in the absence of noise. However, it has the drawback of noise enhancement when $\mathbf{T}$ is ill-conditioned. A remedy is to use a regularized decorrelating matched filter given by

$$\mathbf{F} = \left(\mathbf{T}^H\mathbf{T} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{T}^H. \tag{9}$$

Such a front end does not eliminate multiaccess interference, and the derivation of the channel estimator is now an approximation. It does improve the performance at low SNR in our simulations.

A simple matched filter $\mathbf{F} = \mathbf{T}^H$ can also be used with the algorithm. This introduces a bias to the estimator, and our simulation shows indeed a performance floor as the SNR increases.

E. Multirate, Multicode, and Multiple Antennas

To facilitate multimedia applications, third-generation wireless systems may employ multirate and multicode CDMA. A multicode system assigns multiple codes to the same user. This is equivalent to the unicode case with multiple users having the same multipath channel. Suppose that a set of codes $\{\mathbf{c}_{in},\, i \in I\}$ is allocated to a particular user with channel $\mathbf{h}$. After decorrelation, the channel and symbol estimation problem reduces to

$$\hat{\mathbf{h}} = \arg\max_{\|\mathbf{g}\|=1} \mathbf{g}^H\left(\sum_{i\in I}\hat{\mathbf{R}}_i\right)\mathbf{g}.$$

This effectively increases the number of samples available for the estimation of the channel. (The same situation occurs for estimating downlink channels if the mobile user has the knowledge of multiple spreading codes.) In WCDMA, for example, in-phase and quadrature components are transmitted with different channelization codes followed by aperiodic spreading. It is without loss of generality to treat the in-phase and quadrature components as signals from two different users with different spreading codes but with the same propagation channel. The spreading codes for the in-phase and quadrature part are known and, therefore, can be used in the channel estimation.

Multirate transmission can be accomplished in several ways by using multiple spreading gains or variable chip rates. In both cases, only the decorrelating matched filter needs to be modified, and the channel estimation and symbol detection algorithm applies directly.

Only minor changes are necessary when the proposed algorithm is applied to multiple receiving antenna systems. The reception at the mth antenna is given by

$$\mathbf{y}^m = \mathbf{T}\mathbf{H}^m\mathbf{s} + \mathbf{w}^m.$$

The same decorrelating matched filter is applied at each antenna element. The channels can then be estimated either separately per antenna element or jointly by exploiting the fact that all have the same symbol sequence. The same rank-one decomposition is used in both cases.

IV. Identifiability

We have so far assumed that the code matrix $\mathbf{T}$ has full column rank and is therefore invertible from the left. This assumption is usually valid for systems with large spreading gains. Under this assumption, it is clear that each user's channel is identifiable up to a scaling factor. A single pilot symbol will be sufficient to remove the scalar ambiguity. Singularity does occur when the spreading gain is small and the system is heavily loaded. Even if the case of having singular $\mathbf{T}$ is rare, it remains of theoretical interest to investigate whether the channel is still identifiable, and if not, how many known symbols are necessary and how to place these known symbols.

We now present an identifiability result that is more general than existing conditions. The condition is independent of the channel parameters and can be checked easily offline, and appropriate measures can be taken if it is not satisfied. More significant, perhaps, is that it decouples the identifiability of a particular user from that of others; one user's channel may be identifiable even when those of others are not. The proof of the following theorem gives the algorithm that identifies the channel when the identifiability condition holds.

Theorem 1: Let $\mathbf{T}_{in}$ be the code matrix of user i for symbol n and $\check{\mathbf{T}}_{in}$ the submatrix of $\mathbf{T}$ after removing $\mathbf{T}_{in}$. The channel $\mathbf{h}_i$ of user i is identifiable if there exists an n such that

$$\mathcal{R}\{\mathbf{T}_{in}\} \bigcap \mathcal{R}\{\check{\mathbf{T}}_{in}\} = \{\mathbf{0}\}. \tag{10}$$

Proof: If (10) holds for some n, then the range space of $\mathbf{T}$ can be decomposed into the sum of two subspaces, i.e., there exists a matrix $\mathbf{V}$ with rank$(\mathbf{T})$ − rank$(\mathbf{T}_{in})$ linearly independent columns such that

$$\mathcal{R}\{[\mathbf{T}_{in} \quad \mathbf{V}]\} = \mathcal{R}\{\mathbf{T}\}.$$

Let $\tilde{\mathbf{T}} \triangleq [\mathbf{T}_{in} \quad \mathbf{V}]$. If $\mathbf{y}$ satisfies (1), and there is no noise

$$\tilde{\mathbf{T}}^\dagger \mathbf{y} = \begin{bmatrix} * \\ \mathbf{h}_i s_{in} \\ * \end{bmatrix}$$

which implies that $\mathbf{h}_i$ is identifiable up to a scaling factor from $\tilde{\mathbf{T}}^\dagger\mathbf{y}$. ■

Because (10) only needs to hold for some n, the use of long codes in CDMA makes the identifiability condition easier to satisfy. For randomly generated codes, the probability of that a channel is not identifiable can be made arbitrarily small by increasing the size of the data block.

Among the few identifiability results for long-code CDMA, Xu and Tsatsanis [10] presented a rank condition of a certain matrix constructed from the spreading codes. The identifiability condition, however, is shown using an asymptotic argument and is not applicable to finite sample cases. The approach by Weiss and Friedlander [12] requires the invertibility of the code matrix after chips containing intersymbol interferences have been removed. If this condition holds, the one presented in Theorem 1 is automatically satisfied but not in the reverse direction.

V. Efficient Implementations

The code matrix $\mathbf{T}$ can be large; a K-user synchronous system with spreading gain of G and L multipath fingers for each user and M symbols in each slot will have a code matrix of size approximately $GM \times MKL$. The complexity of directly inverting $\mathbf{T}$ is of order $GM^3K^2L^2$, and the complexity of applying the inverse $\mathbf{T}^\dagger$ to a vector is of order GM^2KL. In contrast, the standard matched-filter front-end $\mathbf{T}^H$ has a complexity of $GMKL$ because only a fraction $1/M$ of all entries of $\mathbf{T}$ are nonzero.

Our goal in this section is to obtain orders of magnitude reduction in computation and storage requirements. This is accomplished by performing the inversion using time-varying state-space techniques.

In summary, for the synchronous case, our method will have a computational complexity of the order GMK^2L^2, which is linear in terms of the number of symbols in the slot and is at the same level of complexity as required for a decorrelating receiver in the short-code case. The reduction of applying the inverse is also substantial. If $\mathbf{T}^\dagger$ has already been obtained, then the amount of computation required to apply $\mathbf{T}^\dagger$ is of the order $GMKL$. The computational complexity can be reduced further by exploiting the Toeplitz structure of the blocks of $\mathbf{T}$.

The complete theory behind the approach taken here is available in [2], and presentation here is focused on the basic concepts applied to the specialized model.

A. State-Space Representation of a Matrix

Consider an input signal $\mathbf{u}$ and output signal $\mathbf{y} = \mathbf{T}\mathbf{u}$, with arbitrary block-partitioning[3]

$$\mathbf{u} = \left[\mathbf{u}_1^T, \ldots, \mathbf{u}_N^T\right]^T, \qquad \mathbf{y} = \left[\mathbf{y}_1^T, \ldots, \mathbf{y}_N^T\right]^T.$$

The partitioning introduces the notion of "time" or a stage in a computational procedure. The blocks do not need to be of equal size, and some can even be empty dimensional, which represents the absence of the corresponding input or output at that point in time. A matrix or vector with a zero dimension is denoted by "•" (see Section I-C).

A time-varying state-space realization of $\mathbf{y} = \mathbf{T}\mathbf{u}$ has the form

$$\begin{cases} \mathbf{x}_{n+1} &= \mathbf{A}_n\mathbf{x}_n + \mathbf{B}_n\mathbf{u}_n \\ \mathbf{y}_n &= \mathbf{C}_n\mathbf{x}_n + \mathbf{D}_n\mathbf{u}_n \end{cases}$$

[3] Note that the partitioning here is not necessarily the same as the one used in (3).

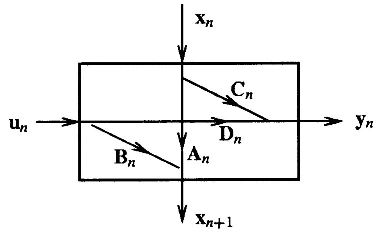


Fig. 3. Time-varying state-space representation at time instant n.

or equivalently[4]

$$\begin{bmatrix}\mathbf{x}_{n+1}\\ \mathbf{y}_n\end{bmatrix} = \mathbf{T}_n \begin{bmatrix}\mathbf{x}_n\\ \mathbf{u}_n\end{bmatrix}, \qquad \mathbf{T}_n \triangleq \begin{bmatrix}\mathbf{A}_n & \mathbf{B}_n\\ \mathbf{C}_n & \mathbf{D}_n\end{bmatrix}$$

where $\mathbf{x}_n$ is a state vector that carries information from one stage of the computation to the next. A graphical representation of this is shown in Fig. 3. The state-space realization specifies a mapping of $\mathbf{u}$ to $\mathbf{y}$, which is necessarily causal: $\mathbf{y}_n$ does not depend on $\mathbf{u}_{n+1}$. It is assumed that the realization starts at time 1 with $\mathbf{x}_1 = \bullet$ (or no state) and ends at time N again with state $\mathbf{x}_{N+1} = \bullet$. Hence, $\mathbf{A}_1 = \bullet$, $\mathbf{A}_N = \bullet$, $\mathbf{C}_1 = \bullet$, and $\mathbf{B}_N = \bullet$.

Consider first an arbitrary $N \times L$ matrix $\mathbf{T}$, with rows $\mathbf{t}_n^H$. A (trivial) realization that models $\mathbf{y} = \mathbf{Tu}$ is obtained by setting $\mathbf{u}_1 = \mathbf{u}, \mathbf{u}_2 = \cdots = \mathbf{u}_N = \bullet$ (i.e., the complete input vector is entered at time 1), and

$$\begin{bmatrix}\mathbf{A}_1 & \mathbf{B}_1\\ \mathbf{C}_1 & \mathbf{D}_1\end{bmatrix} = \begin{bmatrix}\bullet & \mathbf{I}\\ \bullet & \mathbf{t}_1^H\end{bmatrix}, \qquad \begin{bmatrix}\mathbf{A}_N & \mathbf{B}_N\\ \mathbf{C}_N & \mathbf{D}_N\end{bmatrix} = \begin{bmatrix}\bullet & \bullet\\ \mathbf{t}_N^H & \bullet\end{bmatrix}$$

$$\begin{bmatrix}\mathbf{A}_n & \mathbf{B}_n\\ \mathbf{C}_n & \mathbf{D}_n\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \bullet\\ \mathbf{t}_n^H & \bullet\end{bmatrix}, \qquad n = 2, \ldots, N-1.$$

The structure of the realization is shown in Fig. 4.

As a second example, let $\mathbf{T} = [\mathbf{T}^{(1)} \ \mathbf{T}^{(2)}]$ be an arbitrary block-partitioned matrix, where $\mathbf{T}^{(1)}$ has realization $\{\mathbf{A}_n^{(1)}, \mathbf{B}_n^{(1)}, \mathbf{C}_n^{(1)}, \mathbf{D}_n^{(1)}\}$, and $\mathbf{T}^{(2)}$ has realization $\{\mathbf{A}_n^{(2)}, \mathbf{B}_n^{(2)}, \mathbf{C}_n^{(2)}, \mathbf{D}_n^{(2)}\}$.

Then, $\mathbf{T}$ has realization

$$\mathbf{T}_n = \left[\begin{array}{cc|cc}\mathbf{A}_n^{(1)} & 0 & \mathbf{B}_n^{(1)} & 0\\ 0 & \mathbf{A}_n^{(2)} & 0 & \mathbf{B}_n^{(2)}\\ \hline \mathbf{C}_n^{(1)} & \mathbf{C}_n^{(2)} & \mathbf{D}_n^{(1)} & \mathbf{D}_n^{(2)}\end{array}\right].$$

The structure of the realization is shown in Fig. 5

The code matrix $\mathbf{T}$ in our case has a block structure as shown in Fig. 2. By combining the two examples, we can represent any code matrix $\mathbf{T}$, irrespective of the processing gains G_i, offsets D_i, channel lengths L_i, and number of symbols M_i (these can be different for each user). The number of state-space time points N is equal to the number of rows of $\mathbf{T}$. For users with equal parameters, $N = GM + L - 1$. Each state-space stage has one (scalar) output. The input vector is partitioned in blocks of L_i entries that enter the system at appropriate time points and are determined by the starting points of the individual code blocks. The state dimension at each time point is (usually) the number of nonzero entries in the corresponding row of $\mathbf{T}$ (fewer if the row contains the start or end of a block).

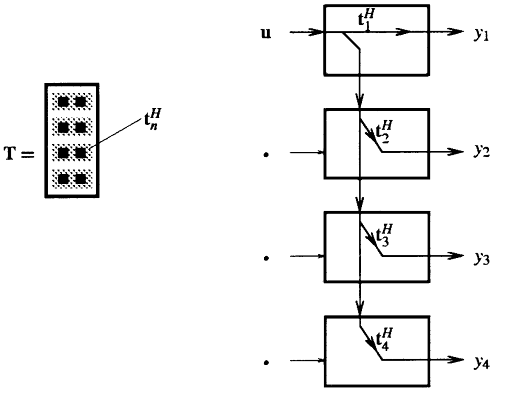


Fig. 4. Computational network for $\mathbf{y} = \mathbf{Tu}$.

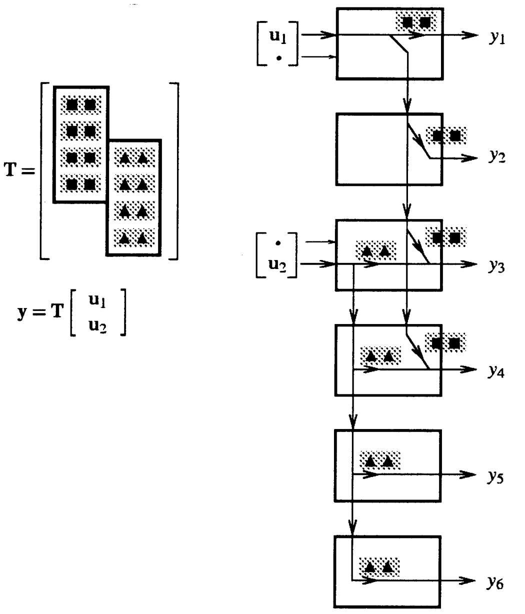

$\mathbf{u}_1$
•
y_1
$\mathbf{T} =$
y_2
•
$\mathbf{u}_2$
y_3
y_4
$\mathbf{y} = \mathbf{T}\begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix}$
y_5
y_6

Fig. 5. Computational network for $\mathbf{T} = [\mathbf{T}^{(1)} \ \mathbf{T}^{(2)}]$.

B. QR Factorization in State-Space

Consider a matrix $\mathbf{T}$ with realization $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}_{n=1}^N$. Assume that $\mathbf{T}$ is "tall" and full column rank ($\mathbf{T}^H\mathbf{T}$ is full rank). To compute the left inverse $\mathbf{T}^\dagger$, our aim is to first compute a QR factorization $\mathbf{T} = \mathbf{QR}$, where $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$ and $\mathbf{R}$ is square and lower triangular, and then to invert each of the factors: $\mathbf{T}^\dagger = \mathbf{R}^{-1}\mathbf{Q}^H$. The computation of the QR factorization can be done in state-space by the following recursion.

For given $\mathbf{Y}_{n+1}$, consider the (economy-size) QR factorization

$$\begin{aligned}\begin{bmatrix}\mathbf{Y}_{n+1}\mathbf{A}_n & \mathbf{Y}_{n+1}\mathbf{B}_n\\ \mathbf{C}_n & \mathbf{D}_n\end{bmatrix} &\triangleq \mathbf{Q}_n\mathbf{L}_n\\ &\triangleq \begin{bmatrix}\mathbf{A}_n^Q & \mathbf{B}_n^Q\\ \mathbf{C}_n^Q & \mathbf{D}_n^Q\end{bmatrix} \cdot \begin{bmatrix}\mathbf{Y}_n & 0\\ \mathbf{C}_n^R & \mathbf{D}_n^R\end{bmatrix}\end{aligned} \tag{11}$$

where $\mathbf{Q}_n$ is "tall" and isometric ($\mathbf{Q}_n^H\mathbf{Q}_n = \mathbf{I}$), and $\mathbf{L}_n$ is lower triangular and full row rank (hence "wide" and possibly in staircase form). Subsequently, partition $\mathbf{L}_n$ as indicated, such that $\mathbf{Y}_n$ and $\mathbf{A}_n$ have the same number of columns, $\mathbf{D}_n^R$ and $\mathbf{D}_n$ have the same number of columns, and $\mathbf{D}_n^R$ has full row rank.

[4] With abuse of notation, we redefine the meaning of $\mathbf{T}_n$ in this section as compared with (1). In this section, n denotes the chip index of the sample vector $\mathbf{y}$.

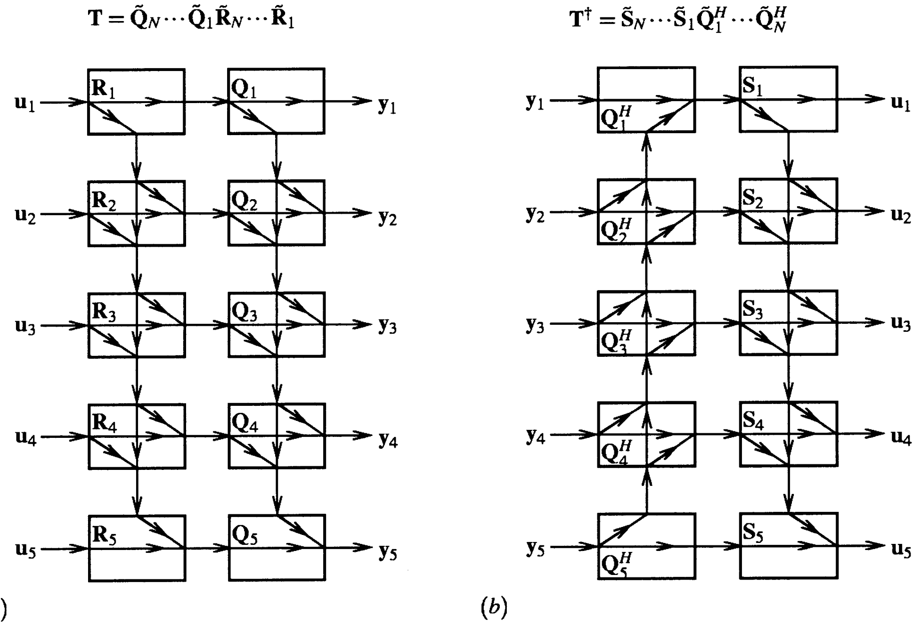


Fig. 6. Inversion. (a) Structure of the QR factorization. (b) Structure of the inverse. Note that the inverse is not causal.

$\mathbf{Q}_n$ is partitioned accordingly. Note that it may happen that $\mathbf{Y}_n$ or $\mathbf{D}_n^R$ is zero dimensional. As a consequence, the indicated “0” entry above $\mathbf{D}_n^R$ may vanish as well.

The recursion is started by $\mathbf{Y}_{N+1} = \bullet$ and is computed for $n = N, N-1, \ldots, 1$ in turn.

Theorem 2: Consider a matrix $\mathbf{T}$ of full column rank, and carry out the recursion (11). Then, $\mathbf{T} = \mathbf{QR}$, where $\mathbf{Q}$ is specified by the realization $\{\mathbf{A}_n^Q, \mathbf{B}_n^Q, \mathbf{C}_n^Q, \mathbf{D}_n^Q\}$ and $\mathbf{R}$ by the realization $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n^R, \mathbf{D}_n^R\}$.

All $\mathbf{D}_n^R$ are square, lower triangular, and invertible. $\mathbf{Q}$ is isometric ($\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$), and $\mathbf{R}$ is lower triangular and invertible.

Proof: See the Appendix.

The structure of the corresponding factorization is shown in Fig. 6(a). In the recursion, the occurrence of a $\mathbf{D}_n^R$, which is not square, indicates that $\mathbf{T}$ is not of full column rank. The recursion can still be continued (and we obtain $\mathbf{T} = \mathbf{QR}$), but $\mathbf{R}$ is not square and invertible but wide.

Note that in our application, $\mathbf{A}_n$ and $\mathbf{B}_n$ are trivial: embeddings of identity matrices of appropriate sizes. Hence, the multiplication by $\mathbf{Y}_{n+1}$ is trivial, and the only actual work in (11) is the QR factorization. Since $\mathbf{Y}_{n+1}$ is triangular and $[\mathbf{C}_n \; \mathbf{D}_n]$ is a single row, the QR factorization can be efficiently implemented as a *QR update* [18].

C. Matrix Inversion via State-Space

Theorem 3: Suppose that $\mathbf{R}$ is a square invertible lower triangular matrix. Then, its inverse is also lower triangular. If $\mathbf{R}$ has state-space realization

$$\mathbf{R}_n = \begin{bmatrix} \mathbf{A}_n^R & \mathbf{B}_n^R \\ \mathbf{C}_n^R & \mathbf{D}_n^R \end{bmatrix}, \qquad n = 1, \ldots, N$$

then $\mathbf{S} \triangleq \mathbf{R}^{-1}$ has state-space realization

$$\mathbf{S}_n = \begin{bmatrix} \mathbf{A}_n^R - \mathbf{B}_n^R\mathbf{D}_n^{R^{-1}}\mathbf{C}_n^R & \mathbf{B}_n^R\mathbf{D}_n^{R^{-1}} \\ -\mathbf{D}_n^{R^{-1}}\mathbf{C}_n^R & \mathbf{D}_n^{R^{-1}} \end{bmatrix}, \qquad n = 1, \ldots, N.$$

Proof: Note that $\mathbf{Ru} = \mathbf{y} \Leftrightarrow \mathbf{Sy} = \mathbf{u}$; hence, $\mathbf{S}$ maps $\mathbf{y}$ to $\mathbf{u}$. Since $\mathbf{S}$ is lower triangular (causal)

$$\begin{cases} \mathbf{x}_{n+1} &= \mathbf{A}_n^R\mathbf{x}_n + \mathbf{B}_n^R\mathbf{u}_n \\ \mathbf{y}_n &= \mathbf{C}_n^R\mathbf{x}_n + \mathbf{D}_n^R\mathbf{u}_n \end{cases}$$
$$\Leftrightarrow$$
$$\begin{cases} \mathbf{x}_{n+1} &= \mathbf{A}_n^R\mathbf{x}_n + \mathbf{B}_n^R(-\mathbf{D}_n^{R^{-1}}\mathbf{C}_n^R\mathbf{x}_n + \mathbf{D}_n^{R^{-1}}\mathbf{y}_n) \\ \mathbf{u}_n &= -\mathbf{D}_n^{R^{-1}}\mathbf{C}_n^R\mathbf{x}_n + \mathbf{D}_n^{R^{-1}}\mathbf{y}_n. \end{cases}$$

Invertibility of $\mathbf{R}$ guarantees that all $\mathbf{D}_n^R$ are square and invertible. ■

Theorem 4: Suppose that $\mathbf{Q}$ is an isometry ($\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$) with realization

$$\mathbf{Q}_n = \begin{bmatrix} \mathbf{A}_n^Q & \mathbf{B}_n^Q \\ \mathbf{C}_n^Q & \mathbf{D}_n^Q \end{bmatrix}, \qquad n = 1, \ldots, N$$

where all $\mathbf{Q}_n$ are isometric. Then, $\mathbf{Q}$ has a left inverse $\mathbf{Q}^H$ with an *anticausal* state-space realization

$$\mathbf{Q}_n^H = \begin{bmatrix} \mathbf{A}_n^{Q^H} & \mathbf{C}_n^{Q^H} \\ \mathbf{B}_n^{Q^H} & \mathbf{D}_n^{Q^H} \end{bmatrix}, \qquad n = 1, \ldots, N. \qquad (12)$$

The anticausal realization in (12) corresponds to the equations (backward recursion)

$$\begin{cases} \mathbf{x}_n &= \mathbf{A}_n^{Q^H}\mathbf{x}_{n+1} + \mathbf{C}_n^{Q^H}\mathbf{u}_n \\ \mathbf{y}_n &= \mathbf{B}_n^{Q^H}\mathbf{x}_{n+1} + \mathbf{D}_n^{Q^H}\mathbf{u}_n \end{cases} \qquad n = N, N-1, \ldots, 1.$$

The state-space realizations of $\mathbf{S}$ and $\mathbf{Q}^H$ are obtained *locally* for every state matrix $\mathbf{R}_n$ and $\mathbf{Q}_n$ independently.

The preceding theorems can be used to invert more general matrices, in particular, the code matrix $\mathbf{T}$. After deriving a state-space realization $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}$ of $\mathbf{T}$, compute the QR factorization $\mathbf{T} = \mathbf{QR}$ in state-space, and invert each of the factors in state-space. This provides an implementation of $\mathbf{T}^\dagger = \mathbf{SQ}^H$ in factored form, although $\mathbf{T}^\dagger$, $\mathbf{R}$, and $\mathbf{Q}$ are never explicitly evaluated. The first factor is causal ($\mathbf{S}$ is lower triangular) and the second anticausal ($\mathbf{Q}^H$ is upper triangular). The structure of the computational network is shown in Fig. 6(b). As is seen from this structure, the "complexity" of $\mathbf{T}$ and $\mathbf{T}^\dagger$ is the same, even if $\mathbf{T}^\dagger$ is a full matrix (i.e., with mixed causality).

D. Complexity

For our application to code matrix inversion, we study the complexity of this solution in more detail. Consider a code matrix $\mathbf{T}$ (refer to Fig. 2). The complexity of directly computing $\mathbf{Tu}$ is on the order of the number of nonzero entries of $\mathbf{T}$. The complexity of computing $\mathbf{Tu}$ in state-space is precisely the same if we do not count the multiplications by $\mathbf{A}_n$, $\mathbf{B}_n$ (since these are embeddings of identity matrices). The storage requirement is also equal to the number of nonzero entries (not counting the identity matrices). For users with equal parameters, this number is on the order of $GMKL$.

We now look at the complexity of the factors of $\mathbf{T} = \mathbf{QR}$ (in state-space). Each factor $\mathbf{R}_n$ has the same or less nonzero entries as the corresponding $\mathbf{T}_n$. The $(\mathbf{A}_n^R, \mathbf{B}_n^R)$ pair is the same as for $\mathbf{T}_n$ (identity matrices) and does not count for the complexity. Storage for $\mathbf{R}$ requires at most the same number of entries as storage for $\mathbf{T}$.

In contrast, each factor $\mathbf{Q}_n$ has full state-space descriptions but can be specified by a small number of (2 × 2) Givens rotations: on the order of the number of nonzero entries of $\mathbf{T}_n$ if the QR factorization (11) is implemented as a QR update [18]. Hence, we recommend storage of $\mathbf{Q}$ in implicit (factorized) form by storing the Givens rotations. The storage requirement is then on the order of nonzero entries of $\mathbf{T}$. The number of nonzero entries of each $\mathbf{T}_n$ is on the order of KL, and there are $N = GM$ stages, so that the complexity of computing the QR factorization is on the order of $GM(KL)^2$.

Multiplication by $\mathbf{Q}_n$ or $\mathbf{Q}_n^H$ can be carried out by applying the corresponding Givens rotations. The complexity of applying $\mathbf{Q}$ or $\mathbf{Q}^H$ to a vector is, hence, the same as the complexity of applying $\mathbf{T}$: on the order of $GM\ KL$ operations. It is in our case not needed to store $\mathbf{Q}$; we can apply the Givens rotations directly to the observation vector $\mathbf{y}$ while they are being computed in the backward QR iteration.

For $\mathbf{S} = \mathbf{R}^{-1}$, we also do not recommend the explicit formation and storage of $\mathbf{S}_n$ since these have full state-space matrices. Rather, $\mathbf{S}_n$ can be applied implicitly by back-substitution from $\mathbf{R}_n$, for $n = 1, \ldots, N$. The complexity and storage for $\mathbf{S}$ is, in this case, the same as for $\mathbf{R}$.

In summary, the complexity of computing the state-space representation of $\mathbf{T}^\dagger$ is on the order of $GM(KL)^2$ operations, which is linear in the number of symbols per user M and comparable with the complexity of a decorrelating receiver in the short-code case. The storage requirement of $\mathbf{T}^\dagger$ in state-space factored form is about two times the number of nonzero entries in $\mathbf{T}$, or order $GMKL$. The complexity of applying $\mathbf{T}^\dagger$ to the observation vector is also order $GMKL$. This is the same as the complexity of applying the matched filter $\mathbf{T}^H$. In contrast, note that $\mathbf{T}^\dagger$ is a full matrix, with GM^2KL entries. Computing $\mathbf{T}^\dagger$ directly requires order $GM^3K^2L^2$ operations, and applying it to a vector requires GM^2KL operations. The benefit in complexity of using state-space representations is thus on the order of M^2 and M, respectively.

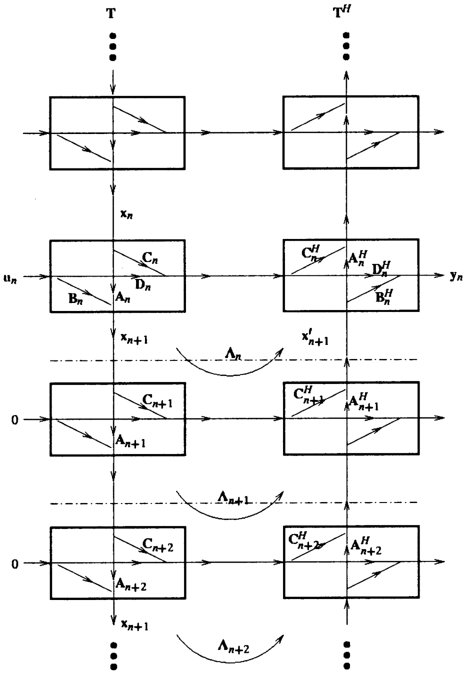


Fig. 7. Computation of $\mathbf{T}^H\mathbf{T}$.

E. Computation of $\mathbf{T}^H\mathbf{T}$ and $\mathbf{T}\mathbf{T}^H$

In the computation of the noise covariance, expressions for $\mathbf{T}^H\mathbf{T}$ and $\mathbf{T}\mathbf{T}^H$ are needed.

Theorem 5: Let $\mathbf{T}$ be a block-lower triangular matrix with state-space realization $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}$. Consider $\mathbf{M} = \mathbf{T}^H\mathbf{T}$. A realization for the lower triangular part of $\mathbf{M}$ is given by

$$\mathbf{M}_n = \begin{bmatrix} \mathbf{A}_n & \mathbf{B}_n \\ \mathbf{B}_n^H\mathbf{\Lambda}_n\mathbf{A}_n + \mathbf{D}_n^H\mathbf{C}_n & \mathbf{B}_n^H\mathbf{\Lambda}_n\mathbf{B}_n + \mathbf{D}_n^H\mathbf{D}_n \end{bmatrix}$$

where $\mathbf{\Lambda}_n$ is specified by the backward recursion

$$\mathbf{\Lambda}_{n-1} = \mathbf{A}_n^H\mathbf{\Lambda}_n\mathbf{A}_n + \mathbf{C}_n^H\mathbf{C}_n, \qquad n = N, N-1, \ldots, 1$$

initialized by $\mathbf{\Lambda}_N = \bullet$. Similarly, a realization for the lower triangular part of $\mathbf{N} = \mathbf{T}\mathbf{T}^H$ is given by

$$\mathbf{N}_n = \begin{bmatrix} \mathbf{A}_n & \mathbf{A}_n\mathbf{\Lambda}_n\mathbf{C}_n^H + \mathbf{B}_n\mathbf{D}_n^H \\ \mathbf{C}_n & \mathbf{C}_n\mathbf{\Lambda}_n\mathbf{C}_n^H + \mathbf{D}_n\mathbf{D}_n^H \end{bmatrix}$$

where $\mathbf{\Lambda}_n$ is now specified by the forward recursion

$$\mathbf{\Lambda}_{n+1} = \mathbf{A}_n\mathbf{\Lambda}_n\mathbf{A}_n^H + \mathbf{B}_n\mathbf{B}_n^H, \qquad n = 1, 2, \ldots, N$$

initialized by $\mathbf{\Lambda}_1 = \bullet$.

Proof: By inspection of Fig. 7, consider the mapping of an input $\mathbf{u}_n$ and a state $\mathbf{x}_n$ to the corresponding output $\mathbf{y}_n$ and new state $\mathbf{x}_{n+1}$,

$$\begin{cases} \mathbf{x}_{n+1} & = \mathbf{A}_n\mathbf{x}_n + \mathbf{B}_n\mathbf{u}_n \\ \mathbf{y}_n & = \left(\mathbf{B}_n^H\mathbf{\Lambda}_n\mathbf{A}_n + \mathbf{D}_n^H\mathbf{C}_n\right)\mathbf{x}_n \\ & \quad + \left(\mathbf{B}_n^H\mathbf{\Lambda}_n\mathbf{B}_n + \mathbf{D}_n^H\mathbf{D}_n\right)\mathbf{u}_n \end{cases}$$

where $\mathbf{\Lambda}_n$ is the transfer of $\mathbf{x}_{n+1}$ to $\mathbf{x}'_{n+1}$. It satisfies

$$\mathbf{\Lambda}_n = \mathbf{A}_{n+1}^H\mathbf{\Lambda}_{n+1}\mathbf{A}_{n+1} + \mathbf{C}_{n+1}^H\mathbf{C}_{n+1}.$$

(A formal proof appears in [2, p. 366].) ∎

The preceding recursions are useful in the computation of the noise covariance after the decorrelating matched filter. If $\mathbf{w}$ is a white noise vector with power normalized to $\sigma^2 = 1$, and $\mathbf{n} = \mathbf{T}^\dagger\mathbf{w} = (\mathbf{T}^H\mathbf{T})^{-1}\mathbf{T}^H\mathbf{w}$, then the covariance of $\mathbf{n}$ is given by

$$\mathbb{E}\left(\mathbf{n}\mathbf{n}^H\right) = \left(\mathbf{T}^H\mathbf{T}\right)^{-1} = \mathbf{S}\mathbf{S}^H$$

where $\mathbf{T} = \mathbf{Q}\mathbf{R}$, and $\mathbf{S} = \mathbf{R}^{-1}$. A state-space realization $\{\mathbf{A}_n^S, \mathbf{B}_n^S, \mathbf{C}_n^S, \mathbf{D}_n^S\}$ for $\mathbf{S}$ was derived before. Thus, Theorem 5 (applied to $\mathbf{S}$) gives a recursion to compute a realization for the lower part of $\mathbf{S}\mathbf{S}^H$. The upper part is simply the transpose.

In the identification algorithm in Section III-C, we are only interested in the main (block)-diagonal of $\mathbb{E}(\mathbf{n}\mathbf{n}^H)$ (the autocovariances of size $L_i \times L_i$). In this case, it suffices to compute

$$\mathbb{E}\left(\mathbf{n}_n\mathbf{n}_n^H\right) = \mathbf{C}_n^S\mathbf{\Lambda}_n\mathbf{C}_n^{S^H} + \mathbf{D}_n^S\mathbf{D}_n^{S^H}$$

where

$$\mathbf{\Lambda}_{n+1} = \mathbf{A}_n^S\mathbf{\Lambda}_n\mathbf{A}_n^{S^H} + \mathbf{B}_n^S\mathbf{B}_n^{S^H}, \qquad n = 1, 2, \ldots, N.$$

F. Computation of the Regularized Front End in State-Space

The approach presented in previous sections can also be used to implement the regularized decorrelating matched filter front end $\mathbf{F}$ in (9). In particular, let $\mathbf{u} = \mathbf{F}\mathbf{y}$. Then

$$\begin{aligned}\mathbf{u} &= \left(\mathbf{T}^H\mathbf{T} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{T}^H\mathbf{y}\\ &= \left(\mathbf{T}^H\mathbf{T} + \sigma^2\mathbf{I}\right)^{-1}\left[\mathbf{T}^H \quad \sigma\mathbf{I}\right]\begin{bmatrix}\mathbf{y}\\ \mathbf{0}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{T}\\ \sigma\mathbf{I}\end{bmatrix}^\dagger\begin{bmatrix}\mathbf{y}\\ \mathbf{0}\end{bmatrix}. \qquad (13)\end{aligned}$$

Thus, if

$$\begin{bmatrix}\mathbf{T}\\ \sigma\mathbf{I}\end{bmatrix} \triangleq \mathbf{M} \triangleq \mathbf{Q}^M\mathbf{R}^M$$

is an economy-size QR factorization for $\mathbf{M}$ (where $\mathbf{R}^M$ is square triangular and $\mathbf{Q}^M$ is tall and isometric), then

$$\mathbf{u} = \left(\mathbf{R}^M\right)^{-1}\left(\mathbf{Q}^M\right)^H\begin{bmatrix}\mathbf{y}\\ \mathbf{0}\end{bmatrix}.$$

The QR factorization and factor inversion can be done in state space, as before. Thus, $\mathbf{u}$ is the output of a computational structure similar to the one in Fig. 6(b). The only new aspect is the derivation of a realization for $\mathbf{M}$. This is simply obtained from the realization of $\mathbf{T}$ by extending the D-matrix by $\sigma\mathbf{I}$:

$$\mathbf{M}_n = \left[\begin{array}{c|c}\mathbf{A}_n & \mathbf{B}_n\\ \hline \mathbf{C}_n & \mathbf{D}_n\\ \mathbf{0} & \sigma\mathbf{I}\end{array}\right], \qquad n = 1, \ldots, N.$$

As an aside, note that after channel estimation, in addition, MMSE-type receivers $\mathbf{F} = (\mathbf{H}^H\mathbf{T}^H\mathbf{T}\mathbf{H} + \sigma^2\mathbf{I})^{-1}\mathbf{H}^H\mathbf{T}^H$ can be constructed in this way, if $\mathbf{T}$ is replaced by $\mathbf{T}\mathbf{H}$. Since $\mathbf{H}$ is block-diagonal, a state-space realization for $\mathbf{T}\mathbf{H}$ is simply derived from that of $\mathbf{T}$. We omit the details.

VI. Performance Analysis

We present here a brief analysis of the proposed algorithm, focusing primarily on the bit error rate (BER) performance of the whitened RAKE receiver. The analysis of the mean square error (MSE) of the channel estimate can also be done using perturbation techniques applied to the dominant eigenvector. Such an approach, however, does not lend itself to insights, and hence, we defer the MSE evaluation to Section VII, where we compare the proposed channel estimator to the Cramér–Rao bound (CRB) via simulations.

There are no existing techniques for the BER analysis for blind multiuser detection of long-code CDMA due to two major obstacles. First, blind detectors are functions of transmitted symbols and noise realizations. The coupling between channel estimate and bit error makes the analysis intractable. Second, existing blind multiuser detectors [7]–[12] usually have complicated operations involving all users and their channels.

The decorrelating matched filter algorithm separates users in a deterministic and channel independent way, which makes the bit error analysis local to each user. Yet, we still need to decouple the transmitted symbols and the noise realization from the blind detector. A reasonable approach is to analyze future errors by first conditioning the analysis on a realization of the channel estimation and evaluate the BER of incoming symbols. This obviously is an approximation that, in our simulation, is shown to be accurate.

A. BER of the Blind Decorrelating Rake Receiver

The BER is, in general, time varying in long-code CDMA. Here, we calculate the bit error probability for binary phase shift keying (BPSK) signaling and average the error rate over time. For user i, conditioned on the estimated channel $\hat{\mathbf{h}}_i$, a whitened RAKE receiver is applied to $\mathbf{u}_{in}$, which is the output of the decorrelating matched filter corresponding to user i. The whitened RAKE detector produces a detected symbol, from (8)

$$\tilde{s}_{in} = \text{sign}\left\{\text{real}\left\{\hat{\mathbf{h}}_i^H\mathbf{\Sigma}_{in}^{-1}\mathbf{u}_{in}\right\}\right\}$$

where $\mathbf{\Sigma}_{in}$ is the $L_i \times L_i$ submatrix obtained from the $(\sum_{j=1}^{i-1} M_j) + n$th diagonal block of $\mathbf{T}^\dagger(\mathbf{T}^\dagger)^H$. For a system using BPSK with noise power spectral density[5] σ^2 and bit energy E_i for the ith user, the conditional bit error probability for the nth symbol is given by

$$\Pr\left(s_{in} \neq \tilde{s}_{in}|\hat{\mathbf{h}}_i\right) = Q\left(\gamma_{in}\sqrt{\frac{2E_i}{\sigma^2}}\right)$$

where $Q(\cdot)$ is the tail function of the Gaussian distribution,[6] and γ_{in} is the loss with respect to the ideal BPSK system

$$\gamma_{in} \triangleq \frac{\text{real}\left\{\hat{\mathbf{h}}_i^H\mathbf{\Sigma}_{in}^{-1}\mathbf{h}_i\right\}}{\sqrt{G\hat{\mathbf{h}}_i^H\mathbf{\Sigma}_{in}^{-1}\hat{\mathbf{h}}_i}}.$$

[5] A standard notation in digital communication is N_0.

[6] $Q(\alpha) \triangleq (1/\sqrt{2\pi})\int_\alpha^\infty e^{-x^2/2}\,dx$.

For perfect orthogonal codes, this reduces to the standard single-user BPSK performance. The average BER for a block of M_i symbols is given by

$$\overline{P}_i = \frac{1}{M_i} \sum_{k=1}^{M_i} \mathbb{E} \left\{ Q \left(\gamma_{in} \sqrt{\frac{2E_i}{\sigma^2}} \right) \right\} \tag{14}$$

where the expectation is taken over all $\hat{\mathbf{h}}$.

B. BER for the Matched Filter Detector

We contrast the performance with the standard matched filter approach that assumes perfectly orthogonal codes. Specifically, given $\hat{\mathbf{h}}_i$ then, to detect bit $\mathbf{s}_{in}$, a matched filter with L_i fingers is first applied to the received signal $\mathbf{y}$

$$\mathbf{r}_{in} = \mathbf{T}_{in}^H \mathbf{y} = \mathbf{T}_{in}^H \mathbf{T} \mathbf{H} \mathbf{s} + \mathbf{T}_{in}^H \mathbf{w}.$$

Next, the output of the matched filter is combined using the estimated $\hat{\mathbf{h}}_i$ to form the detection statistic

$$z_{in} = \text{real} \left\{ \hat{\mathbf{h}}_i^H \mathbf{r}_{in} \right\} = \alpha_{in} s_{in} + b_{in}(\mathbf{s}, \hat{\mathbf{h}}_i, \mathbf{h}) + \eta_{in}$$

where

$$\begin{aligned} \alpha_{in} &= \text{real} \left\{ \hat{\mathbf{h}}_i^H \mathbf{T}_{in}^H \mathbf{T}_{in} \mathbf{h}_i \right\} \\ b_{in}(\mathbf{s}, \hat{\mathbf{h}}_i, \mathbf{h}) &= \text{real} \left\{ \sum_{n \neq k \text{ or } m \neq i} \hat{\mathbf{h}}_i^H \mathbf{T}_{in}^H \mathbf{T}_{mk} \mathbf{h}_m s_{mk} \right\} \\ \eta_{in} &= \text{real} \left\{ \hat{\mathbf{h}}_i^H \mathbf{T}_{in}^H \mathbf{w} \right\}. \end{aligned}$$

$b_{in}(\mathbf{s}, \hat{\mathbf{h}}_i, \mathbf{h})$ contains the intersymbol and multiaccess interference, and η_{in} contains the noise distributed as

$$\eta_{in} \sim \mathcal{N}(0, \hat{\sigma}_{in}^2), \qquad \hat{\sigma}_{in}^2 \triangleq \tfrac{1}{2} \sigma^2 \hat{\mathbf{h}}_i^H \mathbf{T}_{in}^H \mathbf{T}_{in} \hat{\mathbf{h}}_i.$$

If the detector assumes that the codes are orthogonal and that the channel estimate is perfect, then the detected symbol is given by $\tilde{s}_{in} = \text{sign}\{z_{in}\}$, and the error probability, conditioned on other symbols and the estimated channel, is given by

$$\begin{aligned} P_{in|\hat{\mathbf{h}}_i, \mathbf{s}} &\triangleq \Pr \left(\tilde{s}_{in} \neq s_{in} | \hat{\mathbf{h}}_i, s_{mk}, m \neq i, k \neq n \right) \\ &= \frac{1}{2} Q \left(\frac{\alpha_{in} \sqrt{E_i/G} + |b_{in}(\mathbf{s}, \hat{\mathbf{h}}_i, \mathbf{h})|}{\hat{\sigma}_{in}} \right) \\ &\quad + \frac{1}{2} Q \left(\frac{\alpha_{in} \sqrt{E_i/G} - |b_{in}(\mathbf{s}, \hat{\mathbf{h}}_i, \mathbf{h})|}{\hat{\sigma}_{in}} \right) \\ P_i &\triangleq \mathbb{E} \left\{ \sum_n P_{in|\hat{\mathbf{h}}_i, \mathbf{s}} \right\} \end{aligned} \tag{15}$$

where the expectation is taken over $\mathbf{s}$ and $\hat{\mathbf{h}}_i$.

VII. Simulation Results

In this section, we present some simulation results. For channel estimation, the MSE is used as the performance indicator, and our estimator is compared with the CRB using Monte Carlo runs. For symbol detection, the BER is estimated using Monte Carlo runs and compared with the analytical results.

We consider the following receiver algorithms.

TRR training-based RAKE receiver that uses a matched filter front-end ($\mathbf{F} = \mathbf{T}^H$) along with a training-based channel estimator;

BRR blind RAKE receiver that uses the proposed blind channel estimator with the matched filter front-end ($\mathbf{F} = \mathbf{T}^H$);

DRR decorrelating RAKE receiver that uses the decorrelating matched filter front-end ($\mathbf{F} = \mathbf{T}^\dagger$) and the proposed blind channel estimator;

RDRR regularized decorrelating RAKE receiver that uses the regularized decorrelating matched filter $\mathbf{F} = (\mathbf{T}^H \mathbf{T} + \sigma^2 \mathbf{I})^{-1} \mathbf{T}^H$.

A. Setup

Because our model is deterministic, the simulations use a fixed channel and fixed spreading codes. Where we evaluate the MSE of the channel estimator, the transmitted symbols are also fixed. In evaluating the BER, channels and spreading codes are fixed, and the transmitted bits are generated randomly in each Monte Carlo run. The performance would vary with different channel and spreading parameters, but the qualitative behavior remains the same in various trials. Specific parameters used in the simulations can be found in [16]. All plots shown in this section are based on 1000 Monte Carlo runs.

We consider cases of two and five asynchronous BPSK users with equal power. The spreading codes are generated according to the WCDMA standard with spreading gain $G = 32$, and the code index was selected randomly for each user. The channel for each user has $L = 3$ fingers. The relative delay offsets for the two user and five user case are $D = [0, 23]$ chips and $D = [0, 17, 3, 8, 23]$ chips, respectively. The slot size is $M = 50$ symbols, and one pilot symbol is included at the beginning of the slot of each user. The pilot symbol is used to remove the scaling ambiguity of the blind estimator. The signal-to-noise ratio (SNR) was defined by E_b/σ^2, where the bit energy $E_b = G||\mathbf{h}||^2 E_c$, E_c the transmitting chip energy, and σ^2 is the chip noise variance (or the noise power spectral density).

B. MSE and Cramér–Rao Bound

Figs. 8 and 9 show the MSE performance for two and five asynchronous users, respectively. The MSE performance has the same trend for both cases. We observe that the methods based on the conventional matched filter front-end (TRR and BRR) have a performance floor caused by multiaccess interference. The decorrelating RAKE receiver (DRR), on the other hand, tracks the CRB. However, the gap of DRR to the CRB increases with the number of users. For the five-user case, the conventional matched filter with the proposed blind channel estimator (BRR) shows a better performance than that of DRR

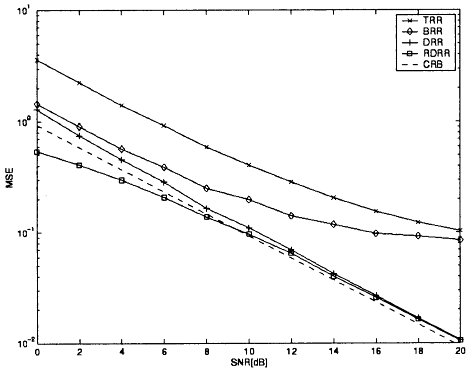


Fig. 8. Channel estimation error (MSE) versus SNR: Two users.

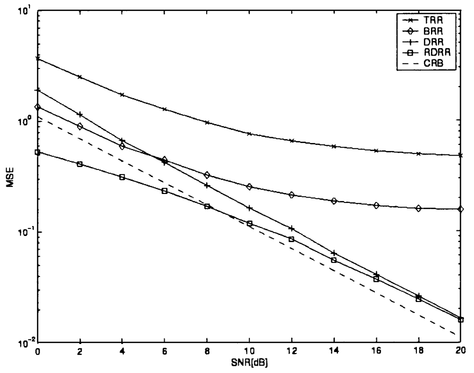


Fig. 9. Channel estimation error (MSE) versus SNR: Five users.

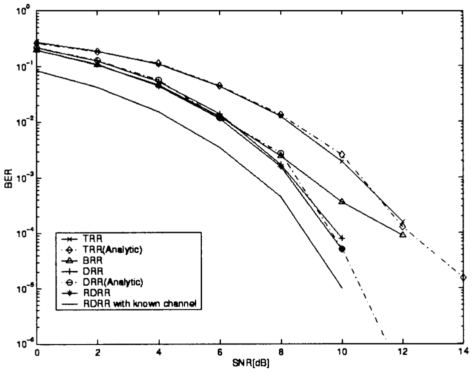


Fig. 10. BER versus SNR: Two users.

at low SNR. The reason is that as the system becomes heavily loaded, the condition number of the code matrix increases, and the decorrelating matched filter enhances the noise. The use of regularized least squares front-end ameliorates this effect, as shown in Figs. 8 and 9. We note that the regularized least squares front-end introduces bias to the estimator, which explains that the MSE of the estimator is lower than the CRB at low SNR.

C. BER Comparison

Fig. 10 shows the average BER performance for the two-user case. The TRR performs worst, especially at high SNR. The performance of DRR is close to that of RDRR because the code matrix in this case is well conditioned. BRR shows a comparable performance at low and medium SNR since it has a reasonably accurate channel estimate, and the correlation with the spreading code ($G = 32$) averages out the other users interference effectively in less-severe multiaccess interference environments. We also observe that the theoretical BER calculations (14) and (15) are close to the performance obtained via Monte Carlo runs, indicating that the assumptions made in Section VI-A are accurate. As benchmarks, we show the performance of receivers with knowledge of the true channel.

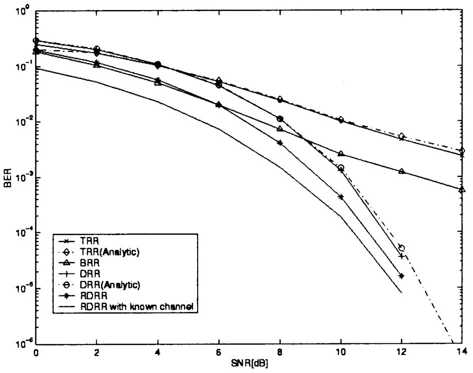


Fig. 11. BER versus SNR: Five users.

Fig. 11 shows the average BER performance for five users. The performance floors of TRR and BRR are expected and caused by multiaccess interference. We observe that RDRR has an appreciable gain over DRR. As in the MSE simulation, BRR performs better than DRR at low SNR due to the noise enhancement effect of DRR. As the SNR increases, however, DRR outperforms the matched filter RAKE.

VIII. Conclusion

In this paper, we considered the problem of channel estimation and symbol detection for long-code CDMA. There are two main contributions. One is a new blind channel estimation and symbol detection algorithm. The technique can be easily amended for semiblind estimation, and it requires a small number of samples. This makes the technique suitable for rapidly fading environments. The proposed approach uses the RAKE structure, which makes it possible to apply our algorithm to a subset of users in a group estimation setting.

The second contribution is an efficient implementation of the decorrelating receiver using time-varying state-space techniques. This part is critical if the decorrelating RAKE is to be used in practice.

The algorithms do not rely on statistical ergodicity nor on synchronization among the users but do assume that the codes of all users are known, as well as their delay offsets. This is often the case in the uplink of a mobile communication system, where there is a separate "finger searcher" that identifies dominant multipaths. It should thus be straightforward to apply the proposed techniques in practical systems.

APPENDIX
PROOF OF THEOREM 2

Lemma 1: Let be given a time-varying realization $\mathbf{T}_n = \{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}$ of $\mathbf{T}$. Then, $\mathbf{T} = \tilde{\mathbf{T}}_N \cdots \tilde{\mathbf{T}}_2\tilde{\mathbf{T}}_1$, where $\tilde{\mathbf{T}}_n$ is an embedding of $\mathbf{T}_n$

$$\tilde{\mathbf{T}}_n \triangleq \left[\begin{array}{c|cccccc} \mathbf{A}_n & & & \mathbf{B}_n & & & \\ \hline & \mathbf{I} & & & & & \\ & & \ddots & & & & \\ & & & \mathbf{I} & & & \\ \mathbf{C}_n & & & \mathbf{D}_n & & & \\ & & & & \mathbf{I} & & \\ & & & & & \ddots & \\ & & & & & & \mathbf{I} \end{array}\right].$$

(There are $n-1$ and $N-n$ identity matrices in the diagonal sequences, respectively.) Moreover, matrix $\mathbf{T}$ is block-lower triangular and has the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{D}_1 & & & \\ \mathbf{C}_2\mathbf{B}_1 & \mathbf{D}_2 & & \\ \vdots & \ddots & \ddots & \\ \mathbf{C}_N\mathbf{A}_{N-1}\cdots\mathbf{A}_2\mathbf{B}_1 & \cdots & \mathbf{C}_N\mathbf{B}_{N-1} & \mathbf{D}_N \end{bmatrix}. \tag{16}$$

Conversely, if a matrix $\mathbf{T}$ has this form, then it has a state-space realization $\mathbf{T}_n = \{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}$.

Proof: The proof is by direct verification by applying the given factorization to the vector $\mathbf{u} = [\bullet|\mathbf{u}_1^T, \mathbf{u}_2^T, \ldots, \mathbf{u}_N^T]^T$ (where $\bullet$ represents $\mathbf{x}_1$) and computing $\mathbf{y} = [\bullet|\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_N^T]^T$ (where $\bullet$ represents $\mathbf{x}_{N+1}$). To verify (16), multiply the factors, and use $\mathbf{A}_1 = \bullet$, $\mathbf{A}_N = \bullet$, $\mathbf{C}_1 = \bullet$, and $\mathbf{B}_N = \bullet$. ■

Proof of Theorem 2: Recall the factorization $\mathbf{T} = \tilde{\mathbf{T}}_N\tilde{\mathbf{T}}_{N-1}\cdots\tilde{\mathbf{T}}_1$, and consider the first factor $\mathbf{T}_N$. Since $\mathbf{A}_N = \bullet$, $\mathbf{B}_N = \bullet$, and $\mathbf{Y}_{N+1} = \bullet$

$$\mathbf{T}_N = \begin{bmatrix} \mathbf{A}_N & \mathbf{B}_N \\ \mathbf{C}_N & \mathbf{D}_N \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{N+1}\mathbf{A}_N & \mathbf{Y}_{N+1}\mathbf{B}_N \\ \mathbf{C}_N & \mathbf{D}_N \end{bmatrix}.$$

The first step in the recursion is the QR factorization

$$\begin{aligned} \mathbf{Q}_N^H\mathbf{T}_N &= \begin{bmatrix} \mathbf{A}_N^Q & \mathbf{B}_N^Q \\ \mathbf{C}_N^Q & \mathbf{D}_N^Q \end{bmatrix}^H \begin{bmatrix} \mathbf{Y}_{N+1}\mathbf{A}_N & \mathbf{Y}_{N+1}\mathbf{B}_N \\ \mathbf{C}_N & \mathbf{D}_N \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{Y}_N & 0 \\ \mathbf{C}_N^R & \mathbf{D}_N^R \end{bmatrix}. \end{aligned}$$

Premultiplying $\mathbf{T}$ by $\tilde{\mathbf{Q}}_N^H$ gives

$$\begin{aligned} \tilde{\mathbf{Q}}_N^H\mathbf{T} &= \left[\begin{array}{c|cc} \mathbf{A}_N^Q & & \mathbf{B}_N^Q \\ \hline & \mathbf{I} & \\ \mathbf{C}_N^Q & & \mathbf{D}_N^Q \end{array}\right]^H \\ &\quad \cdot \left[\begin{array}{c|cc} \mathbf{Y}_{N+1}\mathbf{A}_N & & \mathbf{Y}_{N+1}\mathbf{B}_N \\ \hline & \mathbf{I} & \\ \mathbf{C}_N & & \mathbf{D}_N \end{array}\right] \tilde{\mathbf{T}}_{N-1}\cdots\tilde{\mathbf{T}}_1 \\ &= \left[\begin{array}{c|cccc} \mathbf{Y}_N & & & & 0 \\ \hline & \mathbf{I} & & & \\ & & \ddots & & \\ & & & \mathbf{I} & \\ \mathbf{C}_N^R & & & & \mathbf{D}_N^R \end{array}\right] \\ &\quad \cdot \left[\begin{array}{c|cccc} \mathbf{A}_{N-1} & & & \mathbf{B}_{N-1} & \\ \hline & \mathbf{I} & & & \\ & & \ddots & & \\ \mathbf{C}_{N-1} & & & \mathbf{D}_{N-1} & \\ & & & & \mathbf{I} \end{array}\right] \tilde{\mathbf{T}}_{N-2}\cdots\tilde{\mathbf{T}}_1 \\ &= \left[\begin{array}{c|cccc} \mathbf{Y}_N\mathbf{A}_{N-1} & & & \mathbf{Y}_N\mathbf{B}_{N-1} & \\ \hline & \mathbf{I} & & & \\ & & \ddots & & \\ \mathbf{C}_{N-1} & & & \mathbf{D}_{N-1} & \\ \mathbf{C}_N^R\mathbf{A}_{N-1} & & & \mathbf{C}_N^R\mathbf{B}_{N-1} & \mathbf{D}_N^R \end{array}\right] \\ &\quad \cdot \tilde{\mathbf{T}}_{N-2}\cdots\tilde{\mathbf{T}}_1. \end{aligned}$$

We subsequently obtain

$$\begin{aligned} &\tilde{\mathbf{Q}}_{N-1}^H\tilde{\mathbf{Q}}_N^H\mathbf{T} \\ &= \left[\begin{array}{c|cccc} \mathbf{Y}_{N-1} & & & 0 & \\ \hline & \mathbf{I} & & & \\ & & \ddots & & \\ \mathbf{C}_{N-1}^R & & & \mathbf{D}_{N-1}^R & \\ \mathbf{C}_N^R\mathbf{A}_{N-1} & & & \mathbf{C}_N^R\mathbf{B}_{N-1} & \mathbf{D}_N^R \end{array}\right] \\ &\quad \cdot \left[\begin{array}{c|cccc} \mathbf{A}_{N-2} & \mathbf{B}_{N-2} & & & \\ \hline & \mathbf{I} & & & \\ \mathbf{C}_{N-2} & \mathbf{D}_{N-2} & & & \\ & & \mathbf{I} & & \\ & & & \mathbf{I} & \end{array}\right] \cdot \tilde{\mathbf{T}}_{N-3}\cdots\tilde{\mathbf{T}}_1 \\ &= \left[\begin{array}{c|ccc} \mathbf{Y}_{N-1}\mathbf{A}_{N-2} & \mathbf{Y}_{N-1}\mathbf{B}_{N-2} & & \\ \hline & \mathbf{I} & & \\ \mathbf{C}_{N-2} & \mathbf{D}_{N-2} & & \\ \mathbf{C}_{N-1}^R\mathbf{A}_{N-2} & \mathbf{C}_{N-1}^R\mathbf{B}_{N-2} & \mathbf{D}_{N-1}^R & \\ \mathbf{C}_N^R\mathbf{A}_{N-1}\mathbf{A}_{N-2} & \mathbf{C}_N^R\mathbf{A}_{N-1}\mathbf{B}_{N-2} & \mathbf{C}_N^R\mathbf{B}_{N-1} & \mathbf{D}_N^R \end{array}\right] \\ &\quad \cdot \tilde{\mathbf{T}}_{N-3}\cdots\tilde{\mathbf{T}}_1. \end{aligned}$$

Following the recursion this way, we finally obtain

$$\tilde{\mathbf{Q}}_1^H \cdots \tilde{\mathbf{Q}}_N^H \mathbf{T} = \left[\begin{array}{c|ccc} \mathbf{Y}_1 & & & \\ \hline \mathbf{C}_1^R & \mathbf{D}_1^R & & \\ \mathbf{C}_2^R\mathbf{A}_1 & \mathbf{C}_2^R\mathbf{B}_1 & \mathbf{D}_2^R & \\ \vdots & \vdots & \ddots & \ddots \\ \mathbf{C}_N^R\mathbf{A}_{N-1}\cdots\mathbf{A}_1 & \mathbf{C}_N^R\mathbf{A}_{N-1}\cdots\mathbf{A}_2\mathbf{B}_1 & \cdots \quad \cdots & \mathbf{D}_N^R \end{array}\right].$$

Note that $\mathbf{A}_1 = \bullet$ so that the first column has zero width. Hence, $\mathbf{Y}_1 = \bullet$ (since the $\mathbf{Y}_k$ are wide), and in addition, the first row has empty dimensions. It follows that

$$\tilde{\mathbf{Q}}_1^H \cdots \tilde{\mathbf{Q}}_N^H \mathbf{T} = \begin{bmatrix} \mathbf{D}_1^R & & & \\ \mathbf{C}_2^R\mathbf{B}_1 & \mathbf{D}_2^R & & \\ \vdots & \ddots & \ddots & \\ \mathbf{C}_N^R\mathbf{A}_{N-1}\cdots\mathbf{A}_2\mathbf{B}_1 & \cdots & \cdots & \mathbf{D}_N^R \end{bmatrix} = \mathbf{R}.$$

This is equal to $\mathbf{Q}^H\mathbf{T} = \mathbf{R}$, where $\mathbf{R}$ is lower triangular. Lemma 1 shows that $\mathbf{R} = \tilde{\mathbf{R}}_N \cdots \tilde{\mathbf{R}}_1$ so that $\mathbf{R}$ has the advertised state-space realization. Since $\mathbf{T}$ is full column rank, all $\mathbf{D}_N^R$ are square and invertible so that $\mathbf{R}$ is square and invertible. $\mathbf{Q}$ is isometric since each of its factors $\mathbf{Q}_n$ is isometric. ■

REFERENCES

[1] S. Verdú, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[2] P. Dewilde and A. van der Veen, *Time-Varying Systems and Computations*. Dordrecht, The Netherlands: Kluwer, 1998.

[3] M. Zoltowski, Y. Chen, and J. Ramos, "Blind 2D RAKE receivers based on space-time adaptive MVDR processing for IS-95 CDMA system," in *Proc. 15th IEEE MILCOM*, Atlanta, GA, Oct. 1996, pp. 618–622.

[4] H. Liu and M. Zoltowski, "Blind equalization in antenna array CDMA systems," *IEEE Trans. Signal Processing*, vol. 45, pp. 161–172, Jan. 1997.

[5] Y. Chen, M. D. Zoltowski, J. Ramos, C. Chatterjee, and V. P. Roychowdhury, "Reduced-dimension blind space-time 2-D RAKE receivers for DS-CDMA communication systems," *IEEE Trans. Signal Processing*, vol. 48, pp. 1521–1536, June 2000.

[6] Z. Yang and X. Wang, "Blind turbo multiuser detection for long-code multipath CDMA," *IEEE Trans. Signal Processing*, vol. 50, pp. 112–125, Jan. 2002.

[7] K. Li and H. Liu, "Channel estimation for DS-CDMA with aperiodic spreading codes," in *Proc. ICASSP*, Mar. 1998, pp. 1535–1538.

[8] M. Torlak, B. Evans, and G. Xu, "Blind estimation of FIR channels in CDMA systems with aperiodic spreading sequences," in *Proc. 31st. Asilomar Conf. Signals, Syst., Comput.*, Monterey, CA, Oct. 1997, pp. 495–499.

[9] N. Sidiropoulos and R. Bro, "User separation in DS-CDMA systems with unknown long PN spreading codes," in *Proc. IEEE-SPS Workshop Signal Process. Advances Wireless Commun.*, Annapolis, MD, May 1999, pp. 194–197.

[10] Z. Xu and M. Tsatsanis, "Blind channel estimation for long-code multiuser CDMA systems," *IEEE Trans. Signal Processing*, vol. 48, pp. 988–1001, Apr. 2000.

[11] C. Escudero, U. Mitra, and D. Slock, "A Toeplitz displacement method for blind multipath estimation for long-code DS/CDMA signals," *IEEE Trans. Signal Processing*, vol. 48, pp. 654–665, Mar. 2001.

[12] A. Weiss and B. Friedlander, "Channel estimation for DS-CDMA downlink with aperiodic spreading codes," *IEEE Trans. Commun.*, vol. 47, pp. 1561–1569, Oct. 1999.

[13] T. P. Krauss, W. J. Hillery, and M. D. Zoltowski, "Downlink specific linear equalization for frequency selective CDMA cellular systems," *J. VLSI Signal Process.*, vol. 30, pp. 143–161, 2002.

[14] C. D. Frank, E. Visotsky, and U. Madhow, "Adaptive interference suppression for the downlink of a direct sequence CDMA system with long spreading sequences," *J. VLSI Signal Process.*, vol. 30, pp. 273–291, 2002.

[15] S. Buzzi and H. Poor, "Channel estimation and multiuser detection in long-code DS/CDMA systems," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1476–1487, Aug. 2001.

[16] L. Tong, A. van der Veen, P. Dewilde, and Y. Sung, "Blind decorrelating rake receiver for long-code WCDMA," Cornell Univ., Ithaca, NY, Tech. Rep. ACSP-02-01, Feb. 2002.

[17] E. de Carvalho and D. Slock, "Semi-blind methods for FIR multichannel estimation," in *Signal Processing Advances in Wireless and Mobile Communications: Trends in Channel Estimation and Equalization*, G. Giannakis, Y. Hua, P. Stoica, and L. Tong, Eds. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[18] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1990.

Lang Tong (S'87–M'91–SM'01) received the B.E. degree from Tsinghua University, Beijing, China, in 1985 and the M.S. and Ph.D. degrees in electrical engineering in 1987 and 1990, respectively, from the University of Notre Dame, Notre Dame, IN.

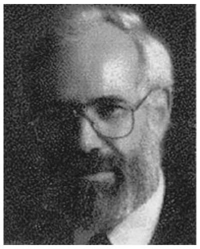
He was a Postdoctoral Research Affiliate at the Information Systems Laboratory, Stanford University, Stanford, CA, in 1991. Currently, he is an Associate Professor with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY. His areas of interest include statistical signal processing, adaptive receiver design for communication systems, signal processing for communication networks, and information theory.

Dr. Tong received the Young Investigator Award from the Office of Naval Research in 1996 and the Outstanding Young Author Award from the IEEE Circuits and Systems Society.

Alle-Jan van der Veen (S'87–M'94–SM'02) was born in The Netherlands in 1966. He graduated (cum laude) from the Department of Electrical Engineering, Delft University of Technology, Delft, The Netherlands, in 1988 and received the Ph.D. degree (cum laude) from the same institute in 1993.

Throughout 1994, he was a postdoctoral scholar with the Scientific Computing/Computational Mathematics Group and the Information Systems Laboratory, Stanford University, Stanford, CA. At present, he is a Full Professor with the Signal Processing Group, Delft Institute of Microelectronics and Submicron Technology (DIMES), Delft University of Technology. His research interests are in the general area of system theory applied to signal processing and, in particular, algebraic methods for array signal processing.

Dr. van der Veen received a 1994 and a 1997 IEEE SPS Young Author paper award and was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1998 to 2001. He is currently chairman of the IEEE SPS SPCOM Technical Committee and Editor-in-Chief of IEEE SIGNAL PROCESSING LETTERS.



Patrick Dewilde (S'69–M'73–SM'81–F'82) received the electrical engineering degree from the University of Leuven, Leuven, Belgium, in 1966, the License in Mathematics from the Belgian Central Examination Commission in 1968, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1970.

He has held various research and teaching positions with the University of California, Berkeley, the University of Lagos, Lagos, Nigeria, and the University of Leuven. In 1977 he became Full Professor of electrical engineering with the Technical University of Delft, Delft, The Netherlands. His research interests included the design of integrated circuits (VLSI) especially in the area of signal processing, large-scale computational problems, theoretical topics in system theory and signal processing, and information management. He has been a project leader of major European projects in microelectronics. The NELSIS design system, which pioneered a unique design information management methodology, was developed under his direction. In 1993, he became the Scientific Director of the Delft Institute of Microelectronics and Submicron Technology (DIMES), which employs more than 300 researchers who are active in advanced microelectronics. He is the author of a large number of scientific publications, a book entitled *Large Scale Modeling of Integrated Circuits* (Dordrecht, The Netherlands: Kluwer, 1988) and one entitled *Time Varying Systems and Computations* (Dordrecht, The Netherlands: Kluwer, 1988).

Dr. Dewilde was elected as a regular member of the Dutch Royal Academy of Science in 1993.

Youngchul Sung (S'91–M'95–S'00) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1993 and 1995, respectively. He is currently with the Adaptive Communications and Signal Processing Laboratory (ACSP), Cornell University, Ithaca, NY, where he is pursuing the Ph.D. degree.

He worked as a research engineer at Goldstar Information and Communications, Ltd., Seoul, from 1995 to 2000. At Goldstar, he developed high data-rate wireless transmission systems and wireless local loop based on wideband code-division multiple-access (CDMA). His areas of interest include communication receiver design, statistical signal processing, and mathematical statistics.