Adaptive Communications and Signal Processing laboratory
Cornell University, Ithaca, NY 14853

Blind Decorrelating RAKE Receiver for Long Code WCDMA

L. Tong, A. van der Veen, P. Dewilde, and Y. Sung

Technical Report No. ACSP-TR-02-01

February 2002

Abstract

The problem of blind and semiblind channel estimation is considered for long code wideband CDMA systems that allow multirate and multicode transmissions. A decorrelating matched filter, implemented efficiently in the state-space, eliminates multiaccess interference and produces a bank of vector processes. Each vector process spans a one-dimensional 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[1].

Index Terms—Long Code Wideband CDMA. Aperiodic spreading sequences/codes. Channel Estimation. Blind and Semiblind Multiuser Detection. Identifiability Conditions. Decorrelating Matched Filter/RAKE Receiver. Fast Algorithms.

EDICS: 3-ACCS (Multiuser and multiaccess communication), 3-CEQU (Channel modeling, estimation, and equalization).

I. Introduction

A. The Problem and the Approach

We consider the problem of joint channel estimation and symbol detection in a long-code wideband 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.

RAKE receivers are widely used in both up-link and down-link 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

[1]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.

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 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 computation complexity and storage requirement. 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 $6,400 \times 3,000$. 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 loose 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 the state space. For cases when the pre-computation of the code matrix is possible, our has the same level of on-line computation and storage requirement as that of the conventional matched filter. If the inversion of code matrix must be performed on-line, we are able to reduce the computation complexity to the same level as that required in the short code case.

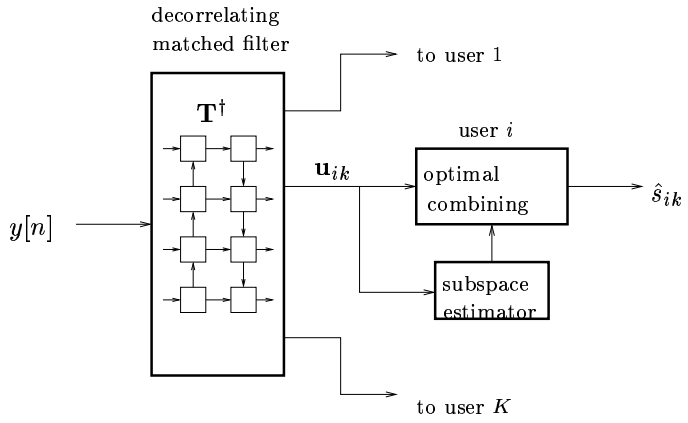


Fig. 1. Receiver structure. The decorrelating matched filter is implemented by an efficient state-space realization.

B. Related Work

A blind (2D) RAKE receiver for long code CDMA was first proposed by Zoltowski, Chen and Ramos in [3] and further developed in [4, 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 2D

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. Also in the literature are second order moment techniques based on the 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 and that some samples can be dropped to eliminate intersymbol interference. 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.

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 mathematical 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 [12,13], hence we will make only a brief derivation here[2].

We assume that K asynchronous users transmit linearly modulated symbols. The transmission is slotted, and user i transmits M_i symbols $\{s_{ik}, k = 1, \cdots, M_i\}$ in each slot. The symbol sequence from user i is represented by the vector $\mathbf{s}_i \triangleq [s_{i1}, \cdots, s_{i,M_i}]^T$. At the transmitter, each symbol is spreaded by an aperiodic code of spreading gain 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}, $\ell = 0, \cdots, L_i$, where h_{ij} can be viewed as the gain of the jth finger of the user i's multipath channel.

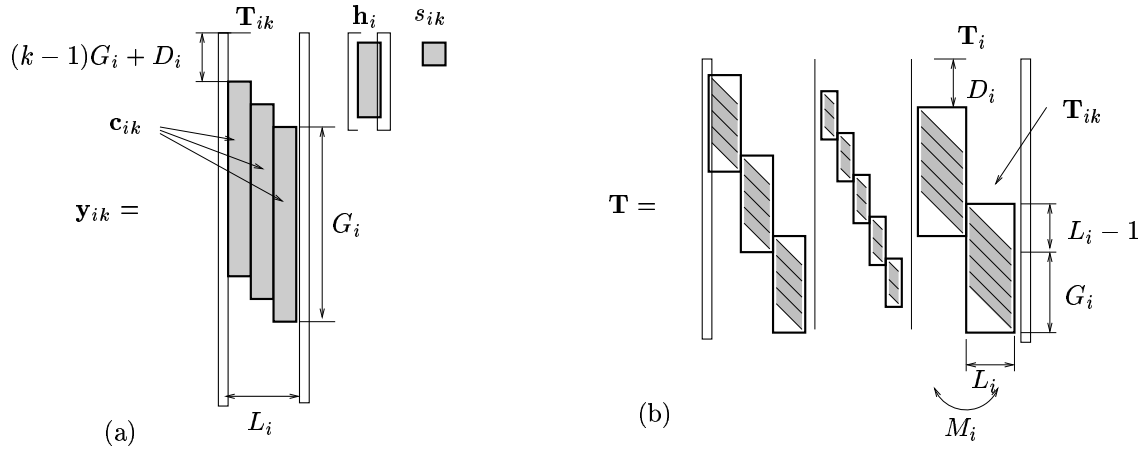


Fig. 2. (a) Structure of $\mathbf{T}_{ik}$; (b) Structure of the code matrix $\mathbf{T}$.

Because the channel is linear, we can first focus on symbol s_{ik} from user i transmitted in the kth symbol interval and set all other symbols and noise to zero. Let the received signal corresponding to symbol s_{ik} be passed through a chip-matched filter and sampled at the chip rate[3]. All samples are put in a vector $\mathbf{y}_{ik}$. As shown in Fig 2, $\mathbf{y}_{ik}$ is made of a linear combination of shifted (delayed) code vector $\mathbf{c}_{ik}$ where $\mathbf{c}_{ik}$ is the segment of G_i chips of user i's spreading code corresponding to the kth symbol. Each shifted code vector is multiplied by the jth fading coefficient h_{ij}, and the received signal corresponding to s_{ik} is given by

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

where $\mathbf{T}_{ik}$ is the code matrix of user i and symbol k (See Fig 2(a)), 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 $\mathbf{T}_{ik}$ corresponds to a discrete multipath. For example, the first column of $\mathbf{T}_{ik}$ is made of $kG_i + D_i$ zeros that model the relative delay of the first path with respect

[2] An algebraic derivation of the model based on Nyquist sampling is given in [14].

[3] In general, sufficient statistics can be obtained by Nyquist sampling. The derivation here will correspond to the even (or odd) subsequence of a $T_c/2$ sampled observation.

to the reference followed by the code vector $\mathbf{c}_{ik}$ and additional zeros that make the size of $\mathbf{y}_{ik}$ the total number of chips of the entire frame. The second column of $\mathbf{T}_{ik}$ models the second multipath 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

$$\begin{aligned} \mathbf{y}_i &= \sum_{k=1}^{M_i} \mathbf{T}_{ik}\mathbf{h}_i s_{ik} = \mathbf{T}_i(\mathbf{I}_{M_i} \otimes \mathbf{h}_i)\mathbf{s}_i \\ \mathbf{T}_i &\triangleq [\mathbf{T}_{i1}, \cdots, \mathbf{T}_{i,M_i}]\,. \end{aligned}$$

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

$$\begin{aligned} \mathbf{y} &= \mathbf{T}\mathbf{H}\mathbf{s} + \mathbf{w} \qquad (1) \\ \mathbf{T} &\triangleq [\mathbf{T}_1, \cdots, \mathbf{T}_K] \\ \mathbf{H} &\triangleq \mathrm{diag}(\mathbf{I}_{M_1} \otimes \mathbf{h}_1, \cdots, \mathbf{I}_{M_K} \otimes \mathbf{h}_K)\,. \end{aligned}$$

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 Figure 2(b). 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 over-parameterize 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, and 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. Given the code matrix, this condition can be verified prior to channel estimation, and it is easy to satisfy for relatively large spreading gains. When (A2) fails, the channel may still be identifiable, and the proposed identification algorithm can be applied with simple modifications. See Sec. IV for details.

III. Blind and Semiblind Decorrelating RAKE

We present in this section the decorrelating RAKE Receiver that jointly estimates channel $\mathbf{h}$ and data $\mathbf{s}$. As illustrated in Fig. 1, we use a decorrelating matched filter $\mathbf{T}^\dagger$ as a front-end to remove mutiaccess interference. Other types of matched filters can also be used, of course, and are briefly discussed in Sec. III-D. We will present the details of an efficient time-varying state-space implementation of the decorrelating matched filter in Sec. 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

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

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

$$\mathbf{u}_{ik} = \mathbf{h}_i s_{ik} + \mathbf{n}_{ik}\,, \quad k = 1, \cdots, M_i\,. \tag{3}$$

Collecting all data for user i gives

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

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

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

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

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

we obtain the least squares estimates

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

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_{ik}.

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 [15].

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 $\mathbf{n}_{ik}$ is colored both in k and in its components. If we ignore the coloring in k, then a simple whitening approach can be applied. We know, from (3), that

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

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

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

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

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

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

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

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

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

Such a front end does not eliminates multiaccess interference, and the 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 $\mathbf{F} = \mathbf{F}^H$ can also be used with the algorithm. This introduces bias to the estimator but with reduced implementation complexity. Our simulation shows a performance floor as 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 $\{\mathbf{c}_{ik}, i \in I\}$ is allocated to a particular user with channel $\mathbf{h}$. After decorrelation, the channel and symbol estimation problem reduces to

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

This effectively increases the number of samples available for the estimation of the channel. (The same situation occurs for estimating down link 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 off-line, 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 $\mathbf{T}_{ik}$ be the code matrix of user i for symbol k, and $\check{\mathbf{T}}_{ik}$ the submatrix of $\mathbf{T}$ after removing $\mathbf{T}_{ik}$. The channel $\mathbf{h}_i$ of user i is identifiable if there exists a k such that

$$\mathcal{R}\{\mathbf{T}_{ik}\} \bigcap \mathcal{R}\{\check{\mathbf{T}}_{ik}\} = \{\mathbf{0}\}. \tag{9}$$

Proof: If (9) holds for some k, 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 $\text{rank}(\mathbf{T}) - \text{rank}(\mathbf{T}_{ik})$ linearly independent columns such that

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

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

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

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

Because (9) only needs to hold for some k, the use of long codes in CDMA makes the identifiability condition easier to satisfy. For randomly generated codes, and large data size M_i, the channel is identifiable almost surely.

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. Another condition is presented by Sidiropoulos and Bro in [9] based on the testing of the k-rank of the channel matrix, the code matrix, and the user signal matrix. This would be channel and signal dependent, and is weaker than that in Theorem 1.

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 packet will have a code matrix of size approximately $MG \times$

MKL. The complexity of inverting $\mathbf{T}$ is of order $GM^3K^2L^2$. The complexity of applying the inverse $\mathbf{T}^\dagger$ is of order GM^2KL. In contrast, the standard matched-filter front-end has the complexity of $MKGL$ because only $1/M$ of all entries of $\mathbf{T}$ are nonzero.

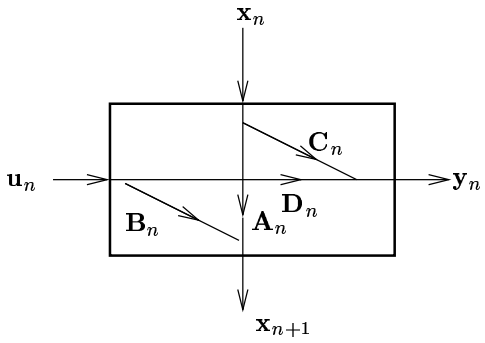


Fig. 3. Time varying state space representation at time instant n.

Our goal in this section is to obtain orders of magnitude reduction in computation and storage requirement. This is accomplished by performing the inversion in state space.

In summary, for the synchronous case, our method will carry the computation complexity of the order $MG(KL)^2$ which is linear in terms of the number of symbols in the packet and is at the same level of complexity as required 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 $MKGL$.

The complete theory behind the approach taken here is available in [2] [16], 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[4]

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

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 "$\bullet$" (see Sec. I-C).

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

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

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

or equivalently[5]

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

where $\mathbf{x}_n$ is a state vector which 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 $\mathbf{A}_1 = \bullet$, $\mathbf{A}_N = \bullet$, $\mathbf{C}_1 = \bullet$, $\mathbf{B}_N = \bullet$.

Consider first an arbitrary $N \times L$ matrix $\mathbf{T}$, with rows $\mathbf{t}_n^H$. A (trivial) realization that models $\mathbf{y} = \mathbf{T}\mathbf{u}$ 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

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{C}_1 & \mathbf{D}_1 \end{bmatrix} = \begin{bmatrix} \bullet & \mathbf{I} \\ \bullet & \mathbf{t}_1^H \end{bmatrix}, \quad \begin{bmatrix} \mathbf{A}_N & \mathbf{B}_N \\ \mathbf{C}_N & \mathbf{D}_N \end{bmatrix} = \begin{bmatrix} \bullet & \bullet \\ \mathbf{t}_N^H & \bullet \end{bmatrix},$$
$$\begin{bmatrix} \mathbf{A}_n & \mathbf{B}_n \\ \mathbf{C}_n & \mathbf{D}_n \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \bullet \\ \mathbf{t}_n^H & \bullet \end{bmatrix}, \quad n = 2, \cdots, N-1$$

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

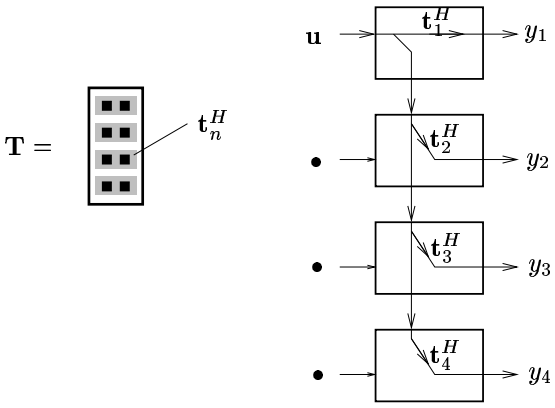


Fig. 4. Computational network for $\mathbf{y} = \mathbf{T}\mathbf{u}$.

As a second example, let $\mathbf{T} = [\mathbf{T}^{(1)} \quad \mathbf{T}^{(2)}]$ 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.

[5]With an abuse of notation, we redefine the meaning of $\mathbf{T}_n$ in this section.

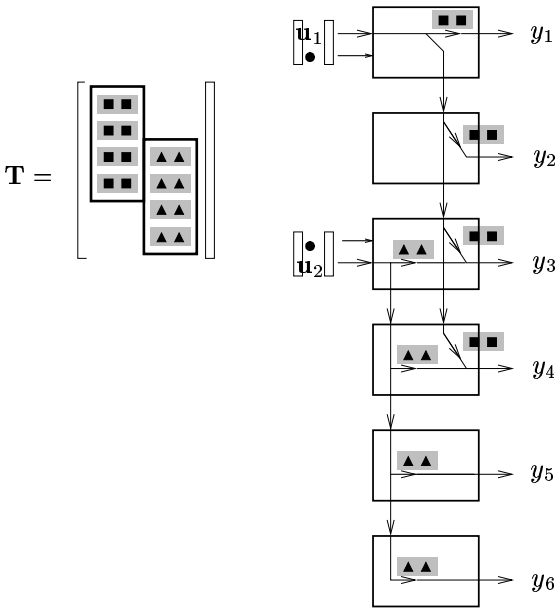


Fig. 5. Computational network for $\mathbf{T} = [\mathbf{T}^{(1)} \;\; \mathbf{T}^{(2)}]$.

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 is equal to the number of rows of $\mathbf{T}$. Each state space stage has 1 (scalar) output. The input vector is partitioned in blocks of L_i entries which enter the system at appropriate time points, 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 $\mathbf{T}$ (less if the row contains the start or end of a block).

B. QR Factorization and Inversion in State Space

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, as is demonstrated by the following theorem.

Theorem 2: Let be given a matrix $\mathbf{T}$ with realization $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}_{n=1,\cdots,N}$. Assume $\mathbf{T}$ is "tall" and has full column rank ($\mathbf{T}^H\mathbf{T}$ is full rank). Let $\mathbf{Y}_{N+1} = \bullet$ and consider the sequence of (economy-size) QR factorizations

$$\begin{bmatrix} \mathbf{Y}_{n+1}\mathbf{A}_n & \mathbf{Y}_{n+1}\mathbf{B}_n \\ \mathbf{C}_n & \mathbf{D}_n \end{bmatrix} \triangleq \underbrace{\begin{bmatrix} \mathbf{A}_n^Q & \mathbf{B}_n^Q \\ \mathbf{C}_n^Q & \mathbf{D}_n^Q \end{bmatrix}}_{\mathbf{Q}_n} \begin{bmatrix} \mathbf{Y}_n & 0 \\ \mathbf{C}_n^R & \mathbf{D}_n^R \end{bmatrix} \tag{10}$$

($n = N, N-1, \cdots, 1$), where $\mathbf{Q}_n$ is "tall" and isometric ($\mathbf{Q}_n^H\mathbf{Q}_n = \mathbf{I}$), and the right factor is lower

triangular (possibly staircase) and partitioned 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 both $\mathbf{Y}_n$ and $\mathbf{D}_n^R$ are full row rank[6].

Then all $\mathbf{D}_n^R$ are square, lower triangular and invertible. Further define the realizations

$$\mathbf{Q}_n = \begin{bmatrix} \mathbf{A}_n^Q & \mathbf{B}_n^Q \\ \mathbf{C}_n^Q & \mathbf{D}_n^Q \end{bmatrix}, \qquad \mathbf{R}_n = \begin{bmatrix} \mathbf{A}_n & \mathbf{B}_n \\ \mathbf{C}_n^R & \mathbf{D}_n^R \end{bmatrix}.$$

Then $\mathbf{T} = \mathbf{QR}$, where $\mathbf{Q}$ is specified by $\mathbf{Q}_n$ and is isometric ($\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$), and $\mathbf{R}$ is specified by $\mathbf{R}_n$ and is lower triangular and invertible.

Proof: See 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 (10) is the QR factorization.

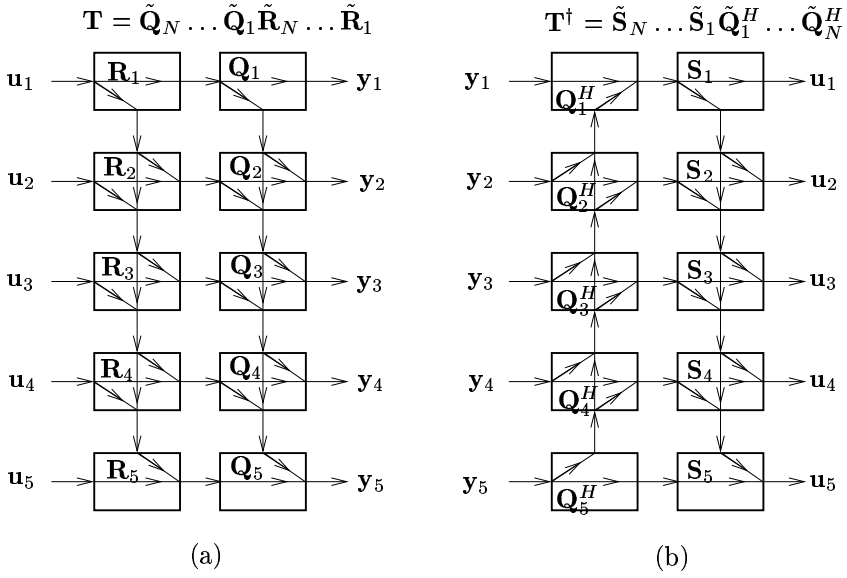


Fig. 6. Inversion. (a) Structure of the QR factorization, (b) structure of the inverse. Note that the inverse is not causal.

C. Matrix Inversion via State Space

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

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

[6]When $\mathbf{D}_n$ or $\mathbf{B}_n$ has zero dimension, so is $\mathbf{D}_n^R$, and the entry "0" above $\mathbf{D}_n^R$ in the recursion is also zero dimension.

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

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

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

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

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

$$\mathbf{Q}_n^H = \begin{bmatrix} \mathbf{A}_n^{Q^H} & \mathbf{C}_n^{Q^H} \\ \mathbf{B}_n^{Q^H} & \mathbf{D}_n^{Q^H} \end{bmatrix}, \qquad n = 1, \cdots, N \tag{11}$$

The anti-causal realization in (11) corresponds to the equations (backward recursion)

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

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), the second anti-causal ($\mathbf{Q}^H$ 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 in the order of the number of non-zero 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 non-zero entries (not counting the identity matrices).

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 non-zero 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: in the order of the number of non-zero entries of $\mathbf{T}_n$. Hence we recommend to store $\mathbf{Q}$ in implicit (factorized) form by storing the Givens rotations. The storage requirement is then in the order of non-zero entries of $\mathbf{T}$. 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 $\mathbf{Q}^H$ is hence the same as the complexity of applying $\mathbf{T}$. It is even possible to apply $\mathbf{Q}_n^H$ directly to the observation vector $\mathbf{y}$ while it is 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 $n = 1, \cdots, N$. The complexity and storage for $\mathbf{S}$ is in this case the same as for $\mathbf{R}$.

In summary, the complexity and storage requirement of $\mathbf{T}^\dagger$ in state-space factored form is about 2 times the number of non-zero entries in $\mathbf{T}$, or $MGLK$ (for users with equal parameters). This is only a factor 2 times the complexity of applying the matched filter $\mathbf{T}^H$. In contrast, note that $\mathbf{T}^\dagger$ is a full matrix, with M^2GLK entries. The benefit in complexity of using state space representations is thus in the order of M, the number of symbols per user per frame.

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

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

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

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

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

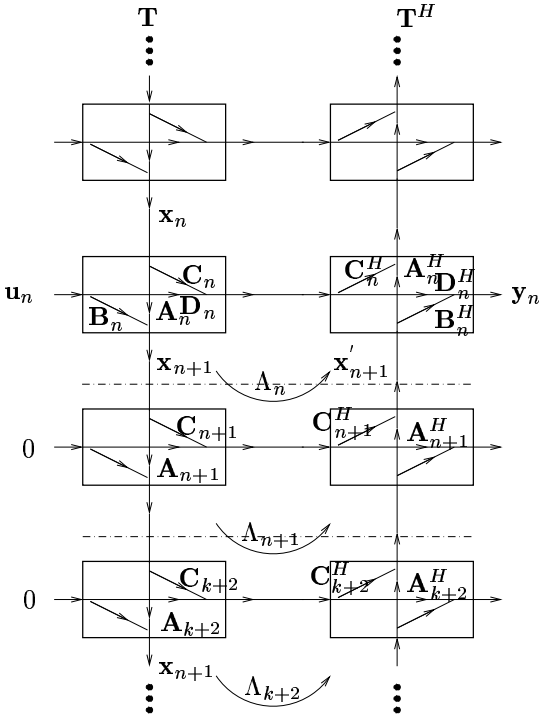


Fig. 7. Computation of $\mathbf{T}^H\mathbf{T}$

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 $\mathbf{x}_{n+1}$.

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

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

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

where $\mathbf{T} = \mathbf{QR}$ 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 $\mathbf{SS}^H$. 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 $\mathbb{E}(\mathbf{nn}^H)$ (the auto-covariances of size $L_i \times L_i$). In this case, it suffices to compute

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

where

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

F. Computation of the MMSE Receiver in State Space

The approach presented in previous sections can also be used to implement the minimum mean square error (MMSE) receiver after the channel has been estimated. The MMSE symbol estimator is given by

$$\begin{aligned} \hat{\mathbf{s}} &= (\mathbf{H}^H \mathbf{T}^H \mathbf{T}\mathbf{H} + \sigma^2 \mathbf{I})^{-1} \mathbf{H}^H \mathbf{T}^H \mathbf{y} \\ &= (\mathbf{H}^H \mathbf{T}^H \mathbf{T}\mathbf{H} + \sigma^2 \mathbf{I})^{-1} \begin{bmatrix} \mathbf{H}^H \mathbf{T}^H & \sigma \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{TH} \\ \sigma \mathbf{I} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \end{aligned} \tag{12}$$

Thus, if

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

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

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

The QR factorization and factor inversion can be done in state space as before. Thus, $\hat{\mathbf{s}}$ 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 $\mathbf{M}$.

A realization $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n\}$ for $\mathbf{T}$ is already known. $\mathbf{H}$ is block-diagonal, with blocks $\mathbf{h}_i$ matching the inputs of $\mathbf{T}$. Define

$$\mathbf{H}_n = \begin{bmatrix} \boldsymbol{\beta}_{1,n} & & \\ & \ddots & \\ & & \boldsymbol{\beta}_{K,n} \end{bmatrix}$$

where

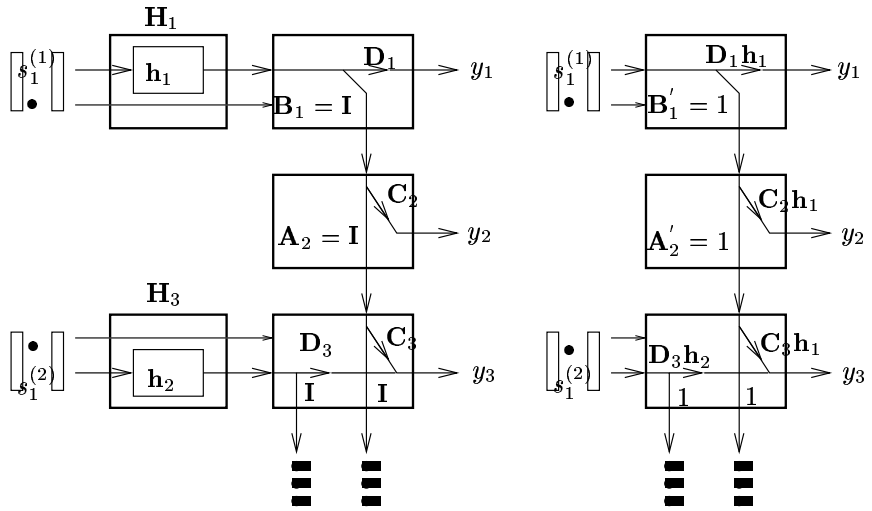
$$\boldsymbol{\beta}_{i,n} = \begin{cases} \mathbf{h}_i, & \mathbf{T} \text{ has an input for user } i \text{ at time } n \\ \bullet, & \text{otherwise.} \end{cases}$$

A realization for $\mathbf{TH}$ is then given by

$$(\mathbf{TH})_n = \begin{bmatrix} \mathbf{A}_n & \mathbf{B}_n\mathbf{H}_n \\ \mathbf{C}_n & \mathbf{D}_n\mathbf{H}_n \end{bmatrix}, \quad n = 1, \cdots, N\,.$$

This is illustrated in Fig. 8(a). Finally, a realization for $\mathbf{M}$ is simply obtained by extending the D-matrix by $\sigma\mathbf{I}$:

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



(a) (b)

Fig. 8. Realization of $\mathbf{TH}$: (a) direct realization, (b) reduced state dimensions.

A few remarks are in order. Since we have already performed a QR-factorization $\mathbf{T} = \mathbf{QR}$, with $\mathbf{R}$ having smaller dimensions than $\mathbf{T}$, we can exploit this. Recall (12). Since $\mathbf{T}^H\mathbf{T} = \mathbf{R}^H\mathbf{R}$, we can write

$$\hat{\mathbf{s}} = (\mathbf{H}^H\mathbf{R}^H\mathbf{R}\mathbf{H} + \sigma^2\mathbf{I})^{-1}\mathbf{H}^H\mathbf{R}^H\mathbf{Q}^H\mathbf{y} = \begin{bmatrix} \mathbf{R}\mathbf{H} \\ \sigma\mathbf{I} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{Q}^H\mathbf{y} \\ \mathbf{0} \end{bmatrix}.$$

Thus define $\mathbf{v} = \mathbf{Q}^H\mathbf{y}$ (it was already computed for the channel estimation step), and use the realization for $\mathbf{R}$ in place of that of $\mathbf{T}$, and $\mathbf{v}$ in place of $\mathbf{y}$.

The shown realization for $\mathbf{TH}$ (or $\mathbf{RH}$) is not minimal. We can reduce the state dimensions from L_i per user-input to 1. This is straightforward to carry out because of the simple structure of $[\mathbf{A}_n,\ \mathbf{B}_n]$: embeddings of $L_i \times L_i$ identity-matrices. Let $[\mathbf{A}'_n,\ \mathbf{B}'_n]$ be a similar-structured matrix, but with embeddings of a scalar 1 in place of each identity matrix. A realization with reduced dimensions is then obtained as

$$(\mathbf{TH})_n = \begin{bmatrix} \mathbf{A}'_n & \mathbf{B}'_n \\ \mathbf{C}_n\mathbf{H}'_n & \mathbf{D}_n\mathbf{H}_n \end{bmatrix}$$

where $\mathbf{H}_n$ is as before, and $\mathbf{H}'_n$ is similar to $\mathbf{H}_n$ but with columns $\mathbf{h}_i$ present at times where the *state* of a user-block is active:

$$\mathbf{H}'_n = \begin{bmatrix} \boldsymbol{\beta}'_{1,n} & & \\ & \ddots & \\ & & \boldsymbol{\beta}'_{K,n} \end{bmatrix}$$

where

$$\boldsymbol{\beta}'_{i,n} = \begin{cases} \mathbf{h}_i, & \mathbf{T}_n \text{ has an input state for user } i \text{ at time } n \\ \bullet, & \text{otherwise}. \end{cases}$$

See Fig. 8(b).

VI. Performance Analysis

We present here a brief analysis of the proposed algorithm, focusing primarily on the bit-error rate 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, hence we defer the MSE evaluation to Sec. VII where we compare the proposed channel estimator to the Cramér-Rao bound (CRB) via simulations.

There are no existing techniques for the bit error rate (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 bit error rate of incoming symbols. This obviously is an approximation which, in our simulation, is shown to be accurate.

A. BER of the Blind Decorrelating Rake Receiver

The bit error rate 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 $\mathbf{u}_{ik}$ which is the output of the decorrelating matched filter corresponding to user i. The whitened RAKE detector produces a detected symbol, from (8),

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

where $\mathbf{\Sigma}_{ik}$ is the $L_i \times L_i$ submatrix obtained from the $(\sum_{j=1}^{i-1} M_j) + k$th diagonal block of $\mathbf{T}^\dagger(\mathbf{T}^\dagger)^H$. For a system using BPSK with noise power spectral density[7] σ^2 and bit energy E_i for the ith user, the conditional bit error probability for the kth symbol is given by

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

where $Q(\cdot)$ is the tail function of the Gaussian distribution[8], and γ_{ik} is the loss with respect to the ideal BPSK system,

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

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

$$\bar{P}_i = \frac{1}{M_i}\sum_{k=1}^{M_i} \mathbb{E}\{Q(\gamma_{ik}\sqrt{\frac{2E_i}{\sigma^2}})\}, \tag{13}$$

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

A.1 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 s_{ik} a matched filter with L_i fingers is first applied to the received signal $\mathbf{y}$,

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

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

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

where

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

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

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

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

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

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

$$\begin{aligned} P_{ik|\hat{\mathbf{h}}_i,\mathbf{s}} &\triangleq \Pr(\tilde{s}_{ik} \neq s_{ik}|\hat{\mathbf{h}}_i, s_{mn}, m \neq i, n \neq k) \\ &= \frac{1}{2}Q(\frac{\alpha_{ik}\sqrt{E_i/G} + |b_{ik}(\mathbf{s},\hat{\mathbf{h}}_i,\mathbf{h})|}{\hat{\sigma}_{ik}}) + \\ &\quad \frac{1}{2}Q(\frac{\alpha_{ik}\sqrt{E_i/G} - |b_{ik}(\mathbf{s},\hat{\mathbf{h}}_i,\mathbf{h})|}{\hat{\sigma}_{ik}}) \\ P_i &\triangleq \mathbb{E}\{\sum_k P_{ik|\hat{\mathbf{h}}_i,\mathbf{s}}\}, \end{aligned} \tag{14}$$

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 was used as the performance indicator, and our estimator was compared with the CRB using Monte Carlo runs. For symbol detection, BER was estimated using Monte Carlo runs and was compared with analytical calculation.

We considered the following scenarios.

TRR: The Training-based Rake Receiver that uses matched filter front-end ($\mathbf{F} = \mathbf{T}^H$) along with training-based channel estimator.

BRR: The Blind RAKE Receiver that uses the proposed blind channel estimator with the matched filter front-end ($\mathbf{F} = \mathbf{T}^H$).

DRR: The Decorrelating RAKE Receiver that uses the decorerlating matched filter front-end ($\mathbf{F} = \mathbf{T}^\dagger$) and the proposed blind channel estimator.

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

A. Setup

Because our model is deterministic, simulations were conducted for a fixed channel and a fixed spreading code. When we evaluated the MSE of the channel estimator, the transmitted symbols were also fixed. In evaluating BER, channels and spreading codes were fixed and the transmitted bits were generated randomly in each Monte Carlo run. The performance would vary with different channel and spreading parameters, but the qualitative behavior remained the same in various trials. Specific parameters used in the simulations can be found in [14]. All plots shown in this section were based on 1000 Monte Carlo runs.

We considered cases of two and five asynchronous BPSK users with equal power. The complex spreading codes were 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 had $L = 3$ fingers. The relative delays for two user and five user case were $D = [0, 23]$ chips and $D = [0, 17, 3, 8, 23]$ chips, respectively. The slot size was $M = 50$, and one pilot symbol was included at the beginning of the slot of each user. The pilot symbol was 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 the chip noise variance (or the noise power spectral density).

B. MSE and Cramér-Rao Bound

Fig. 9 shows the MSE performance for two and five asynchronous users, respectively. The MSE performance had the same trend for both cases. We observe that the methods based on the conventional matched filter front-end (TRR and BRR) had performance floors 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 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 Fig. 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 CRB at low SNR.

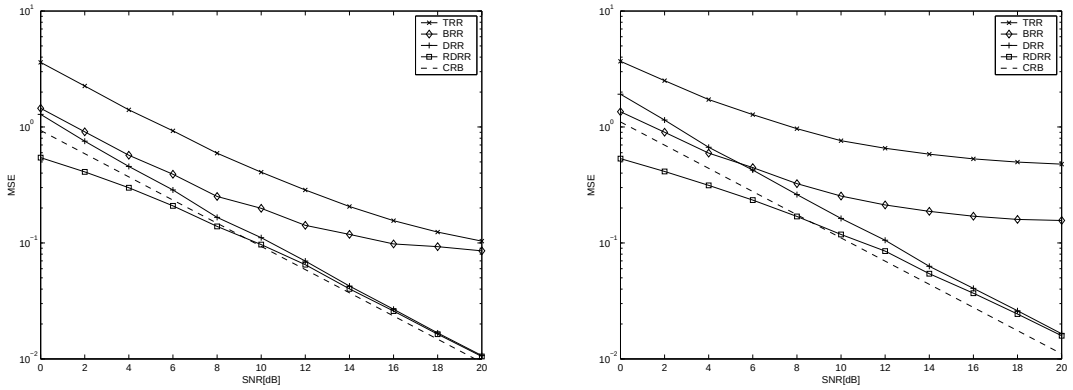


Fig. 9. Channel estimation error (MSE) vs. SNR -(a) 2 Users;(b) 5 Users.

C. Bit Error Rate Comparison

Fig. 10 shows the average BER performance for the two user case. The TRR performed worse, especially at high SNR. The performance of DRR is close to to that of RDRR because the code matrix in this case is well conditioned. BRR showed a comparable performance at low and medium SNR since it has a reasonably accurate channel estimate, and the averaging over spreading gain $G = 32$ mitigates the other user interference effectively in less severe multiaccess interference environment.

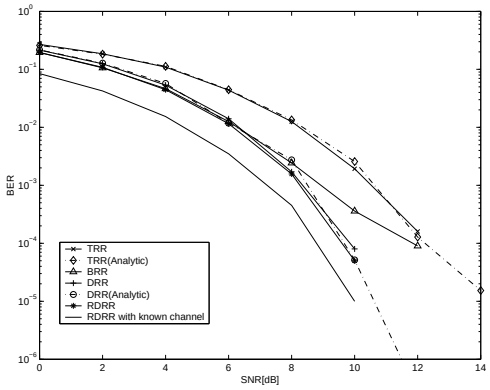


Fig. 10. BER vs. SNR - 2 Users

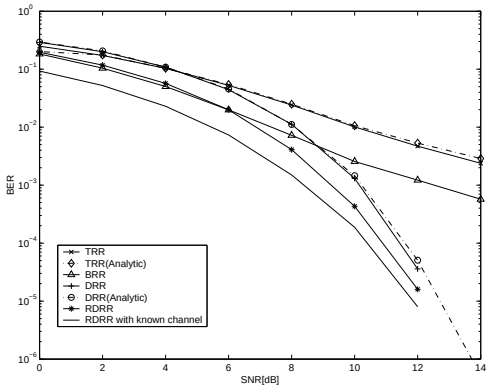


Fig. 11. BER vs. SNR - 5 Users

We also observed that the the theoretical BER calculations (13,14) were close to the performance obtained via Monte Carlo runs, indicating that the assumptions made in Sec. VI-A are accurate. We have included, as benchmarks, cases when the true channel was used in the detectors.

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 observed that RDRR has an appreciable gain over DRR. As in the MSE simulation, BRR performed better than DRR at low SNR due to the effect of noise enhancement of DRR. As the SNR increases, however, DRR outperforms the matched filter RAKE.

D. Average Performance over Different Channel Realizations

In the previous section, the performance of the proposed algorithm was evaluated at a specific channel. The performance variation due to different channel realization was also assessed. The channel was generated independently with Gaussian distribution with three multipaths. All other parameters were the same as in subsection VII-A. Fig. 12 and fig. 13 show the average MSE and BER performance for the two user and five user cases, respectively. The average was taken over 50 different channel realizations. As shown in the figure, the behavior remains the same as a single fixed channel which shows the robustness

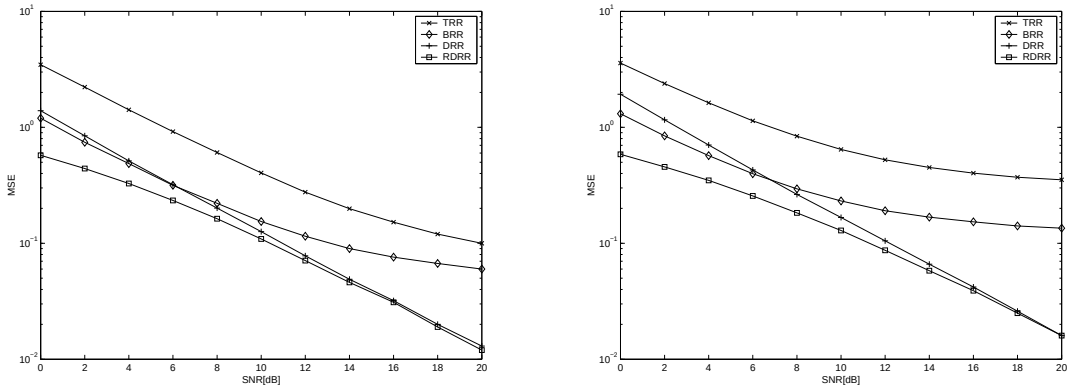


Fig. 12. Average MSE -Left: 2 Users, Right: 5 Users.

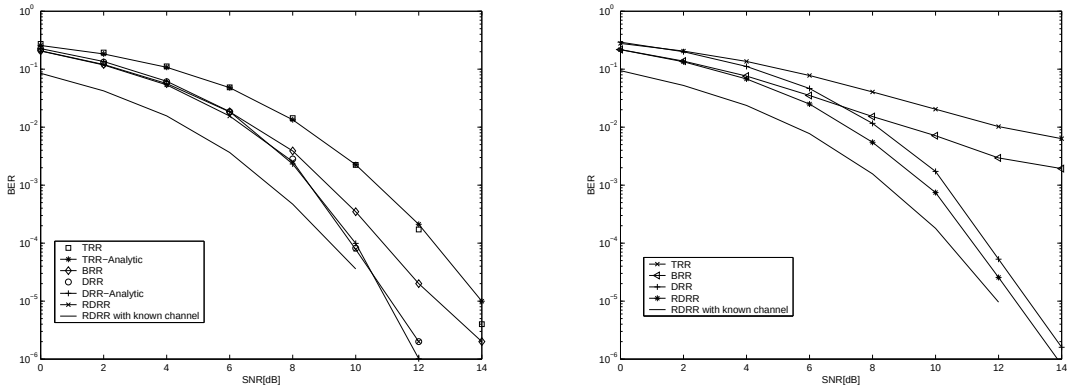


Fig. 13. Average BER -Left: 2 Users, Rigth: 5 Users.

of the proposed algorithm to a random channel realization.

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 amount of samples. This makes the technique suitable for the rapid fading environment. 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 matched filter and MMSE 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 1: General Channel Model

We give a derivation of a general model for long code CDMA based on Nyquist sampling. We first focus on the case when there is one user and a single receiving antenna element and then generalize to the multiuser model with multiple antenna elements.

Suppose that user i has an aperiodic scrambling code $c_i[n]$ with the chip interval index n and transmits a sequence of data symbols $s_i[k]$ with the symbol interval index k.

The transmitted baseband signal is given by

$$s_i(t) = \sum_n d_i[n]\phi(t - nT_c) \tag{15}$$

where T_c is the chip interval, $\phi(t)$ is the chip waveform, and d_n the sequence obtained from the scrambling code and the data sequence. Specifically,

$$d_i[n] \triangleq s_i[\lfloor \frac{n}{G_i} \rfloor]c_i[n] \tag{16}$$

where G_i is the spreading gain for user i, and $\lfloor \cdot \rfloor$ is the floor function. The received signal is given by

$$y_i(t) = \int \tilde{h}_i(\tau)s_i(t - \tau_i - \tau)d\tau + w(t) = \sum_n d_i[n]h_i(t - nT_c) + w(t) \tag{17}$$

where $\tilde{h}_i(t)$ [9] is the baseband propagation channel impulse response, $h_i(t)$ the convolution of the propagation channel response with the chip waveform, τ_i the relative delay with respect to the time reference at the receiver, and $w(t)$ is the white Gaussian noise.

No information will be lost if $y_i(t)$ is sampled at the Nyquist rate. Because the chip waveform has the bandwidth no greater than $2/T_c$, we can sample $y_i(t)$ at the rate twice of the chip rate, which means that there are two samples per chip interval and the received discrete-time sequence can be splitted into even and odd sequences. In particular,

$$y_i^1[n] \triangleq y_i(nT_c) = \sum_l d_i[n - l]h_i^1[l] + w^1[n] \tag{18}$$

$$y_i^2[n] \triangleq y_i(nT_c + \frac{T_c}{2}) = \sum_l d_i[n - l]h_i^2[l] + w^2[n] \tag{19}$$

where $h_i^m[n],\ m = 1, 2$ are two chip-rate sampled channel responses defined as $h_i^1[n] \triangleq h_i(nT_c)$, $h_i^2[n] \triangleq h_i(nT_c + \frac{T_c}{2})$, and $w^m[n],\ m = 1, 2$ is i.i.d. Gaussian noise defined similarly.

[9]The frequency response of the channel can be assumed,without loss of generality, to have the bandwidth no greater than that of the chip waveform.

Assume that $h_i^m[n]$ has only finite support,*i.e.*, there are only L none zero entries. Focusing on user i and writing in the vector-matrix form, we have

$$\mathbf{y}_i^1 = \mathbf{T}(\mathcal{C}_i)(I_{M_i} \otimes \mathbf{h}_i^1)\mathbf{s}_i + \mathbf{w}^1 \tag{20}$$

$$\mathbf{y}_i^2 = \mathbf{T}(\mathcal{C}_i)(I_{M_i} \otimes \mathbf{h}_i^2)\mathbf{s}_i + \mathbf{w}^2 \tag{21}$$

where $\mathbf{h}_i^m = [h_i^m[1], \ldots, h_i^m[L]]^T,\ m = 1, 2$, $\mathbf{w}^1$,$\mathbf{w}^2$ are uncorrelated Gaussian vectors with zero mean and covariance I, and $T(\mathcal{C}_i)$ is defined as in eq.(**??**).

Including all users, we have the following matrix model

$$\mathbf{y}^1 = \sum_{i=1}^{K} \mathbf{T}(\mathcal{C}_i)(I_{M_i} \otimes \mathbf{h}_i^1)\mathbf{s}_i + \mathbf{w}^1 \tag{22}$$

$$= \mathbf{T}(\mathcal{C})diag(I_{M_1} \otimes \mathbf{h}_1^1, \ldots, I_{M_K} \otimes \mathbf{h}_K^1)\mathbf{s} + \mathbf{w}^1 \tag{23}$$

$$= \mathbf{T}(\mathcal{C})\mathcal{D}(\mathbf{h}^1)\mathbf{s} + \mathbf{w}^1 \tag{24}$$

$$\mathbf{y}^2 = \mathbf{T}(\mathcal{C})\mathcal{D}(\mathbf{h}^2)\mathbf{s} + \mathbf{w}^2 \tag{25}$$

The multiple receiving antenna model is derived similarly.

Appendix 2: 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 $\mathbf{T}_n$,

$$\tilde{\mathbf{T}}_n \triangleq \left[\begin{array}{c|ccccc} \mathbf{A}_n & \multicolumn{5}{c}{\mathbf{B}_n} \\ \hline & \mathbf{I} & & & & \\ & & \ddots & & & \\ & & & \mathbf{I} & & \\ \mathbf{C}_n & & & & \mathbf{D}_n & \\ & & & & & \mathbf{I} \\ & & & & & & \ddots \\ & & & & & & & \mathbf{I} \end{array}\right]$$

(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

$$\mathbf{T} = \begin{bmatrix} \mathbf{D}_1 & & & \\ \mathbf{C}_2\mathbf{B}_1 & \mathbf{D}_2 & & \\ \vdots & \ddots & \ddots & \\ \mathbf{C}_N\mathbf{A}_{N-1}\cdots\mathbf{A}_2\mathbf{B}_1 & \cdots & \mathbf{C}_N\mathbf{B}_{N-1} & \mathbf{D}_N \end{bmatrix}. \tag{26}$$

Proof: Direct verification, by applying the given factorization to the vector $\mathbf{u} = \left[\bullet \,|\, \mathbf{u}_1^T,\ \mathbf{u}_2^T, \cdots, \mathbf{u}_N^T\right]^T$ (where $\bullet$ represents $\mathbf{x}_1$), and computing $\mathbf{y} = \left[\bullet \,|\, \mathbf{y}_1^T,\ \mathbf{y}_2^T, \cdots, \mathbf{y}_N^T\right]^T$ (where $\bullet$ represents $\mathbf{x}_{N+1}$). To verify (26), multiply the factors and use $\mathbf{A}_1 = \bullet$, $\mathbf{A}_N = \bullet$, $\mathbf{C}_1 = \bullet$, $\mathbf{B}_N = \bullet$. ■

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

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

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

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

We subsequently obtain

$$\tilde{\mathbf{Q}}_{N-1}^H \tilde{\mathbf{Q}}_N^H \mathbf{T} =$$

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

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

Following the recursion this way, we finally obtain

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

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 also the first row has empty dimensions. It follows that

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

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, UK: Cambridge University Press, 1998.

[2] P. Dewilde and A. van der Veen, *Time-Varying Systems and Computations.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 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 *Proceedings of the 15th IEEE MILCOM*, (Atlanta, GA), pp. 618–622, Oct 1996.

[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, 10 2002.

[7] K. Li and H. Liu, "Channel estimation for DS-CDMA with aperiodic spreading codes," in *Proc. 1999 ICASSP*, pp. 2535–1538, Mar 1998.

[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. Sig. Systems and Computers*, (Monterey, CA), pp. 495–499, Oct 1997.

[9] N. Sidiropoulos and R. Bro, "User separation in DS-CDMA Systems with unknown Long PN Spreading Codes," in *Proc. IEEE-SPS Workshop on Signal Processing Advances in Wireless Communications (SPAWC99)*, (Annapolis, MD.), pp. 194–197, May 1999.

[10] Z. Xu and M. Tsatanis, "Blind channel estimation for long code multiuser CDMA systems," *IEEE Trans.Signal Processing*, vol. SP-48, pp. 988–1001, April 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. SP-48, pp. 654–665, March 2001.

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

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

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

[15] E. de Carvalho and D. Slock, "Semi-blind Methods for FIR Multichannel Estimation," *Signal Processing Advances in Wireless & Mobile Communications: Trends in Channel Estimation and Equalization*, Edited by G. Giannakis, Y. Hua, P. Stoica and L. Tong, Prentice Hall, 2001.

[16] G. Golub and C. V. Loan, *Matrix Computations.* Baltimore, Maryland: The Johns Hopkins University Press, 1990.