Signal Processing 92 (2012) 977–988

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Adaptive BLAST-type decision-feedback equalizers for DS-CDMA systems

Constantinos Rizogiannis [a], Eleftherios Kofidis [b,*], Athanasios A. Rontogiannis [c], Sergios Theodoridis [a,♣]

[a] Department of Informatics and Telecommunications, University of Athens, Panepistimioupolis, 157 84 Athens, Greece
[b] Department of Statistics and Insurance Science, University of Piraeus, 185 34 Piraeus, Greece
[c] Institute for Space Applications and Remote Sensing, National Observatory of Athens, 152 36 P. Penteli, Greece

ARTICLE INFO

Article history:
Received 28 April 2011
Received in revised form
9 October 2011
Accepted 14 October 2011
Available online 25 October 2011

Keywords:
BLAST
Decision feedback equalization (DFE)
Direct sequence code division multiple access (DS-CDMA)
RAKE receiver
Successive interference cancellation (SIC)
Recursive least squares (RLS)

ABSTRACT

In this paper, we propose two new adaptive equalization algorithms for direct sequence code division multiple access (DS-CDMA) systems operating over time-varying and frequency selective channels. The equalization schemes consist of a number of serially connected stages and detect users in an ordered manner, applying a decision feedback equalizer (DFE) at each stage. Both the equalizer filters and the order in which the users are extracted are updated in a recursive least squares (RLS) manner, efficiently realized through time- and order-update recursions. V-BLAST detection ordering is implemented, that is, the stronger signal is extracted first so that the weaker users can be more easily detected. The spreading codes are unavailable at the receiver of the first scheme, whereas the second algorithm employs the RAKE receiver concept, incorporating knowledge of the spreading sequences to offer performance improvement. The bit error rate (BER) performance of the equalizers is evaluated via simulations, in both mild and severe near–far environments. Their superiority over existing techniques is demonstrated.

1. Introduction

In a direct sequence-code division multiple access (DS-CDMA) communication system, several users access the same channel using a common carrier frequency. A unique code sequence is assigned to each user, to spread the signal prior to transmission [20]. It is desirable that code sequences be orthogonal so that the users can be effectively separated at the receiver, eliminating inter-user interference. However, in realistic environments, asynchronous transmission and/or frequency selective channels result in time-shifted and scaled versions of users' code sequences, destroying orthogonality. Multiple access interference (MAI) and inter-symbol interference (ISI) are consequently generated. Moreover, the near–far effect can be quite severe in DS-CDMA systems. This problem comes up when some users are located closer to the receiver compared to others. Hence, signals from distant users are attenuated more than those from users close to the receiver. This greatly affects the detection performance of the weaker users, leading to overall system performance degradation. In a mobile environment, fading is also present and hence the receiver needs to have an adaptation capability as well [9].

It is well-known [16] that when the users are received with different signal strengths, successive interference cancellation (SIC) [8] is the preferable solution, as it exhibits better near–far resistance. SIC consists of a cascade

* Corresponding author. Tel.: +30 210 414 2475; fax: +30 210 414 2340.
E-mail addresses: krizog@di.uoa.gr (C. Rizogiannis), kofidis@unipi.gr (E. Kofidis), tronto@space.noa.gr (A.A. Rontogiannis), stheodor@di.uoa.gr (S. Theodoridis).
♣ Member, EURASIP.

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.10.010

of stages, detecting one user at each stage. The stronger users are detected first and the detected signal is cancelled from the received signal so that the remaining users "see" less MAI. On the other hand, parallel interference cancellation (PIC) is preferred when all users have equal strength at the receiver [15]. An adaptive method performing successive interference cancellation in time-varying environments was proposed in [2], for additive white Gaussian noise (AWGN) and slow Rayleigh fading channels in a near–far situation, and shown to exhibit superior performance compared to the conventional SIC (CSIC) [16] receiver. An improved adaptive SIC (ASIC) technique was developed for AWGN as well as flat Rayleigh fading channels [13]. Its extension to frequency selective Rayleigh fading channels was presented in [12], where its superiority over the CSIC algorithm as well as a PIC method was demonstrated. Multiuser DS-CDMA detection schemes relying on the well-known V-BLAST architecture for multiple input multiple output (MIMO) systems [4] have also been reported. In [19], BLAST is used in the uplink of a DS-CDMA system where users are organized in groups and the same spreading code is assigned to all users of a group. Two schemes which combine multicode CDMA with BLAST architecture in order to achieve high data rates and combat MAI were proposed in [21,6], for uplink and downlink, respectively. All three schemes consider users with multiple antennas while [19,6] also assume perfect channel estimation at the receiver and perfect power control.

In this paper, we propose two new adaptive equalization algorithms of the SIC type for single-antenna DS-CDMA systems, over time-varying and frequency selective channels. Their development relies on the formulation of a DS-CDMA system as a MIMO one and the adoption of existing adaptive solutions of the BLAST-type for MIMO channel equalization [3,18,10]. The proposed equalization schemes assume no channel estimates. They consist of a number of serially connected stages and detect users in an ordered manner, applying a decision feedback equalizer (DFE) at each stage. Both the equalizer filters and the order in which the users are extracted are updated in a recursive least squares (RLS) manner, efficiently realized through time- and order-update recursions. V-BLAST [4] detection ordering is followed, that is, the stronger signal is extracted first, so that the weaker users can be more easily detected. The receiver in the first scheme does not need to know the spreading codes, whereas in the second algorithm the RAKE receiver concept is used, incorporating knowledge of the users' spreading sequences. The BER performance of the proposed equalizers is evaluated via simulations in mild and severe near–far environments. Their superiority compared to ASIC and conventional adaptive techniques is demonstrated. An analysis of their computational complexity is also provided, including possible savings from simplifications like fixed (and not adaptive) ordering.

The rest of the paper is organized as follows. Section 2 describes the system model. The RAKE receiver is briefly summarized in Section 3. The proposed adaptive schemes are presented in Section 4. Section 6 evaluates their performance via simulation results. Conclusions are drawn in Section 7.

Notation. In the following, $(\cdot)^T$ and $(\cdot)^H$ denote transpose and Hermitian transpose of a matrix, respectively. $\boldsymbol{I}_m$ is the mth-order identity matrix, while $\mathbf{0}_{m\times n}$ denotes the $m\times n$ matrix of all zeros. We are using the Matlab notation to designate a submatrix of a given matrix. Complex conjugation is denoted by *. $\otimes$ is the (left) Kronecker product.

2. System model

We consider the uplink of a symbol-synchronous DS-CDMA system with a spreading factor of P chips per symbol, K single-antenna users, and a single-antenna receiver. The users transmit independently the symbol sequences $\boldsymbol{s}_i(k)$, $i=1,2,\ldots,K$. Each such sequence is spread through a P-periodic spreading code $\boldsymbol{c}_i=[c_i(0)\ c_i(1)\ \cdots\ c_i(P-1)]^T$, as shown in Fig. 1. The transmission is through frequency selective channels that vary at the symbol rate, with impulse responses $\boldsymbol{h}_i(k)=[h_{i,0}(k)\ h_{i,1}(k)\ \cdots\ h_{i,L-1}(k)]^T$, $i=1,2,\ldots,K$, of length $L\le P$.

The discrete-time signal x at the receiver's front-end is a summation of signals from all K users, corrupted by additive white Gaussian noise (AWGN). As the aim in this paper is to develop adaptive equalization algorithms for DS-CDMA using earlier work on equalization of MIMO systems, it will be convenient to view the DS-CDMA system as such. Sampling at chip rate and collecting in a $P\times 1$ vector the P successive measurements of x associated with a symbol period, a MIMO formulation with K inputs and P outputs [22,11] results for the DS-CDMA system. Indeed, with the channels being constant during each symbol interval, we get the following input–output relationship:

$$\boldsymbol{x}(k)=\begin{bmatrix} x(kP) \\ x(kP+1) \\ \vdots \\ x(kP+P-1)\end{bmatrix}=\sum_{i=1}^{K}A_i[s_i(k-1)\mathcal{C}_i^{\mathrm{p}}+s_i(k)\mathcal{C}_i^{\mathrm{c}}]\boldsymbol{h}_i(k)+\boldsymbol{v}(k), \tag{1}$$

or equivalently, using $\mathcal{Z}$-transform formulation,

$$\boldsymbol{x}(k)=[A_1(z^{-1}\mathcal{C}_1^{\mathrm{p}}+\mathcal{C}_1^{\mathrm{c}})\boldsymbol{h}_1(k)\cdots A_K(z^{-1}\mathcal{C}_K^{\mathrm{p}}+\mathcal{C}_K^{\mathrm{c}})\boldsymbol{h}_K(k)]\begin{bmatrix} s_1(k)\\ \vdots \\ s_K(k)\end{bmatrix}+\boldsymbol{v}(k), \tag{2}$$

where A_i is the amplitude of the ith user's signal, $s_i(k-1)$ and $s_i(k)$ are the input symbols of the ith user at time instances $k-1$ and k, respectively, $\boldsymbol{v}(k)\in\mathbb{C}^P\times 1$ is additive zero mean, white, complex Gaussian noise vector, and

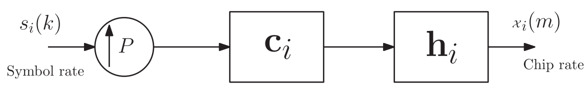

Fig. 1. Block diagram of a single-user DS-CDMA system with spreading factor P.

the $P\times L$ matrices $\mathcal{C}_i^{\rm p}$, $\mathcal{C}_i^{\rm c}$ are built from the $(P+L-1)\times L$ convolution matrix

$$\mathcal{C}_i=\begin{bmatrix} c_i(0) & & \mathbf{0}_{(L-1)\times 1}\\ \vdots & \ddots & c_i(0)\\ c_i(P-1) & & \vdots\\ \mathbf{0}_{(L-1)\times 1} & \ddots & c_i(P-1)\end{bmatrix} \quad (3)$$

as

$$\mathcal{C}_i^{\rm p}=\begin{bmatrix}\mathcal{C}_i(P+1:P+L-1,:)\\ \mathbf{0}_{(P-L+1)\times L}\end{bmatrix},$$

$$\mathcal{C}_i^{\rm c}=\mathcal{C}_i(1:P,:).$$

The latter matrices are associated with the *p*revious and *c*urrent symbols, respectively.

Similarly with Eq. (1), collect $P+L-1$ successive samples of x (instead of P) corresponding to a symbol period followed by the first $L-1$ samples of the next symbol period. Then an alternative MIMO formulation, with K inputs and $P+L-1$ outputs, results, as follows:

$$\begin{aligned}\bar{\boldsymbol{x}}(k)=&\overbrace{[x(kP)\ x(kP+1)\ \cdots\ x(kP+P-1)}^{\boldsymbol{x}^T(k)}\\ &x((k+1)P)\ x((k+1)P+1)\ \cdots\ x((k+1)P+L-2)]^T\\ =&\sum_{i=1}^{K}A_i[s_i(k-1)\overline{\mathcal{C}}_i^{\rm p}\boldsymbol{h}_i(k)+s_i(k)\overline{\mathcal{C}}_i^{\rm c}\boldsymbol{h}_i(k)\\ &+s_i(k)\overline{\mathcal{C}}_i^{\rm cn}\boldsymbol{h}_i(k+1)+s_i(k+1)\overline{\mathcal{C}}_i^{\rm n}\boldsymbol{h}_i(k+1)]+\bar{\boldsymbol{v}}(k),\end{aligned} \quad (4)$$

where

$$\overline{\mathcal{C}}_i^{\rm p}=\begin{bmatrix}\mathcal{C}_i(P+1:P+L-1,:)\\ \mathbf{0}_{P\times L}\end{bmatrix},$$

$$\overline{\mathcal{C}}_i^{\rm c}=\begin{bmatrix}\mathcal{C}_i(1:P,:)\\ \mathbf{0}_{(L-1)\times L}\end{bmatrix},$$

$$\overline{\mathcal{C}}_i^{\rm cn}=\begin{bmatrix}\mathbf{0}_{P\times L}\\ \mathcal{C}_i(P+1:P+L-1,:)\end{bmatrix},$$

$$\overline{\mathcal{C}}_i^{\rm n}=\begin{bmatrix}\mathbf{0}_{P\times L}\\ \mathcal{C}_i(1:L-1,:)\end{bmatrix},$$

and $\bar{\boldsymbol{v}}(k)$ is defined similarly with $\boldsymbol{v}(k)$. Note that the last $L-1$ elements of $\bar{\boldsymbol{x}}(k)$ equal the first $L-1$ elements of $\bar{\boldsymbol{x}}(k+1)$. The matrices $\overline{\mathcal{C}}_i^{\rm p},\overline{\mathcal{C}}_i^{\rm n}$ are associated with the contributions of the preceding $(s_i(k-1))$ and following $(s_i(k+1))$ symbols, respectively. The contribution of the current symbol, $s_i(k)$, is associated with the matrices $\overline{\mathcal{C}}_i^{\rm c}$ and $\overline{\mathcal{C}}_i^{\rm cn}$. Observe that, in a slowly fading scenario, i.e., when $\boldsymbol{h}_i(k+1)\approx\boldsymbol{h}_i(k)$, the corresponding term becomes $A_is_i(k)(\overline{\mathcal{C}}_i^{\rm c}+\overline{\mathcal{C}}_i^{\rm cn})\boldsymbol{h}_i(k)$, where it should be noted that $\overline{\mathcal{C}}_i^{\rm c}+\overline{\mathcal{C}}_i^{\rm cn}=\mathcal{C}_i$. This is a key observation in developing the second equalization algorithm, in Section 4.2, through the incorporation of the RAKE concept.

3. The RAKE receiver

A RAKE receiver [20] utilizes multiple correlators, called *fingers*, to separately detect the signal components transmitted through the L channel paths. At each finger, the desired user's code sequence is correlated with a time-shifted version of the received signal, as shown in Fig. 2. The shifts between adjacent fingers differ by one chip. Finally, the outputs of the L correlators are linearly combined to generate an estimate of the desired user's transmitted symbol. Equivalently, to estimate the symbol of the ith $(i=1,2,\ldots,K)$ user at time k, the RAKE receiver premultiplies[1] Eq. (4) with the $L\times(P+L-1)$ Toeplitz matrix

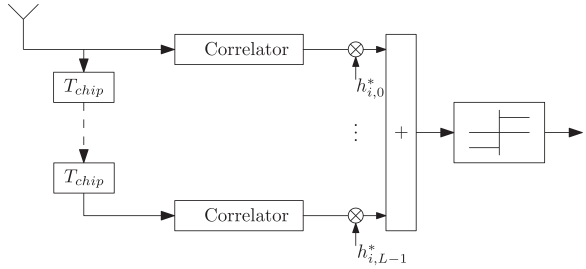

Fig. 2. Block diagram of a single user RAKE receiver with L fingers. Linear combination of the correlators outputs, using the conjugate channel coefficients.

$$\mathcal{C}_i^{\rm RAKE}=\begin{bmatrix} c_i(0) & \cdots & c_i(P-1) & \cdots & 0\\ \vdots & \ddots & & \ddots & \vdots\\ 0 & \cdots & c_i(0) & \cdots & c_i(P-1)\end{bmatrix}=\mathcal{C}_i^T, \quad (5)$$

which contains time-shifted versions of the ith user code sequence $\boldsymbol{c}_i$ in its rows, and linearly combines the elements of the resulting $L\times 1$ vector with the conjugate of the corresponding channel impulse response estimate.

Notice that one can detect all K users by applying a multiuser RAKE receiver.[2] It is realized by premultiplying Eq. (4) with $\mathcal{C}_i^{\rm RAKE}$ for all $i=1,2,\ldots,K$. This concept will be used in Section 4.2 in conjunction with an adaptive BLAST-type MIMO method to derive a new adaptive equalization algorithm.

4. BLAST-type DFE schemes for time-varying channels

4.1. The SR-MUD algorithm

V-BLAST is a well-known architecture for MIMO equalization [4], based on successive detection and cancellation of input streams. Knowledge of the channel is necessary in V-BLAST, for intersymbol interference nulling and symbol detection ordering. Hence, in a time-varying environment, where continuous channel tracking is required, an increase of the system's complexity is entailed.

Based on the equivalence between V-BLAST and generalized DFE (GDFE) [5] methods, an adaptive MIMO DFE

[1] Due to the non-orthogonality of the multipath-distorted spreading codes, this matrix product results in nonzero crosscorrelation values between different code sequences as well as nonzero autocorrelation values between time shifted versions of the same code, giving rise to MAI and ISI effects, respectively.

[2] The multiuser RAKE receiver [20] is based on the use of a single receive antenna for the detection of all users. Knowledge of the code sequences of all users is required.

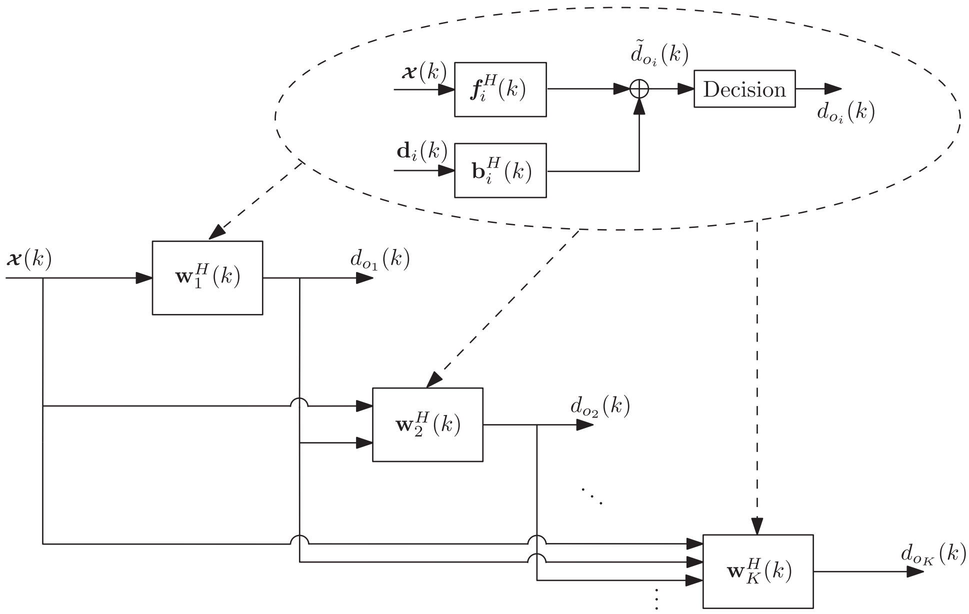

Fig. 3. Block diagram of the adaptive DFE equalizer with successive interference cancellation.

detection scheme with variable detection order[3] was proposed in [3] for flat time-varying channels. Unlike in V-BLAST, *the equalization task does not rely on the availability and update of a channel estimate*. It was shown in [3] that this technique performs similarly to the V-BLAST algorithm with RLS channel tracking, but at a reduced computational complexity. Making use of the MIMO formulation of a DS-CDMA system as presented in (1), a similar equalization scheme can result for a DS-CDMA system with K users and spreading factor P, as shown in Fig. 3, for the flat fading case. At each time instant, the receiver carries out the equalization in K serially connected stages. The users are detected in an ordered manner, applying a DFE at each stage. The stronger users, which are affected less by MAI, are detected first, allowing easier detection for the weaker users.

Let $\{o_1(k),\ldots,o_K(k)\}$, $o_i(k)\in\{1,\ldots,K\}$ be the order in which the users are detected at time k and $U_i(k)=\{1,2,\ldots,K\}-\{o_1(k),o_2(k),\ldots,o_{i-1}(k)\}$ the set of candidate users for stage i, that is the users that have not been detected yet. Although the ordering of users depends on time k, we will skip this notation for the sake of simplicity. Thus, henceforth, o_i will denote the user assigned to the ith stage at time k, unless otherwise stated. As it is shown in Fig. 3, the $(P+i-1)$-dimensional weight vector $\boldsymbol{w}_i(k)$ of the ith stage DFE is composed of the P-dimensional feedforward filter $\boldsymbol{f}_i(k)$ and the $(i-1)$-dimensional feedback filter $\boldsymbol{b}_i(k)$. The input of the feedforward filter is the received vector $\boldsymbol{x}(k)$ as described in (1), while the input of the feedback filter is the vector $\boldsymbol{d}_i(k)=[d_{o_1}(k)\; d_{o_2}(k)\;\cdots\; d_{o_{i-1}}(k)]^T$, which contains the decisions from all the previous stages so that the effect of the already detected users be eliminated. Hence, the weight vector and the input of the ith stage DFE can be written in the form

$$\boldsymbol{w}_i(k)=\begin{cases}\boldsymbol{f}_i(k), & i=1,\\ [\boldsymbol{f}_i^T(k)\;\; \boldsymbol{b}_i^T(k)]^T, & i=2,3,\ldots,K,\end{cases} \tag{6}$$

$$\boldsymbol{y}_i(k)=\begin{cases}\boldsymbol{x}(k), & i=1,\\ [\boldsymbol{x}^T(k)\;\; \boldsymbol{d}_i^T(k)]^T, & i=2,3\ldots,K.\end{cases} \tag{7}$$

The equalizer filters $\boldsymbol{w}_i(k)$ and the order of detection are updated at each stage[4] by minimizing the following set of LS cost functions for all candidate users

$$\mathcal{E}_{i,j}(k)=\sum_{l=1}^{k}\lambda^{k-l}\left|d_j(l)-\boldsymbol{w}_{i,j}^H(k)\boldsymbol{y}_i(l)\right|^2,\quad j\in U_i(k), \tag{8}$$

where $0<\lambda\le 1$ is the forgetting factor and $\boldsymbol{w}_{i,j}(k)$ the equalizer corresponding to the ith stage and the jth user. The minimization of (8) is performed with respect to $\boldsymbol{w}_{i,j}(k)$ and is known [7] to result in

$$\boldsymbol{w}_{i,j}(k)=\boldsymbol{\Phi}_i^{-1}(k)\boldsymbol{z}_{i,j}(k), \tag{9}$$

where $\boldsymbol{\Phi}_i(k)$ stands for the $(P+i-1)\times(P+i-1)$ exponentially time-averaged input autocorrelation matrix, and $\boldsymbol{z}_{i,j}(k)$ for the $(P+i-1)\times 1$ crosscorrelation vector, defined as

$$\boldsymbol{\Phi}_i(k)=\sum_{l=1}^{k}\lambda^{k-l}\boldsymbol{y}_i(l)\boldsymbol{y}_i^H(l), \tag{10}$$

$$\boldsymbol{z}_{i,j}(k)=\sum_{l=1}^{k}\lambda^{k-l}\boldsymbol{y}_i(l)d_j^*(l). \tag{11}$$

[3] The case of a fixed detection order is also covered by the scheme of [3].

[4] Different ordering methods were examined in [17] for an asynchronous CDMA system under AWGN and Rayleigh fading channels. It was shown that, for a linear SIC receiver, ordering after each cancellation and at each symbol is the best ordering method in terms of BER performance.

Table 1
The SR-MUD algorithm.

Initialization: For $i=1,\ldots,K$, $o_i=i$, $\boldsymbol{t}_i(0)=\mathbf{0}$, $\mathcal{E}_i(0)=0$. For $j=1,\ldots,K$, $\boldsymbol{t}_{1,j}(0)=\mathbf{0}$, $\boldsymbol{R}_1^{-1}(0)=\delta^{-1}\boldsymbol{I}$, where δ is a small positive constant.

1. Time update the inverse Cholesky factor $\boldsymbol{R}_1^{-1}(k-1)$ and compute $\boldsymbol{g}_1(k)=\boldsymbol{R}_1^{-H}(k-1)\boldsymbol{y}_1(k)$.
2. $U_1(k)=\{1,2,\ldots,K\}$. For $i=1,2,\ldots,K$
   (a) Order update $\boldsymbol{g}_i(k)$.
   (b) Calculate $d_{o_i}(k)$ by deciding on $\boldsymbol{t}_i^H(k-1)\boldsymbol{g}_i(k)$.
3. For $j=1,2,\ldots,K$
   (a) Time-update $\boldsymbol{t}_{1,j}(k)$.
   (b) Evaluate $\mathcal{E}_{1,j}(k)$.
4. Set $o_1=\arg\min_{j\in U_1(k)}\mathcal{E}_{1,j}(k)$, $\mathcal{E}_1(k)=\mathcal{E}_{1,o_1}(k)$, $\boldsymbol{t}_1(k)=\boldsymbol{t}_{1,o_1}(k)$ and $U_2(k)=U_1(k)-\{o_1\}$.
5. For $i=2,3,\ldots,K$
   (a) For $j\in U_i(k)$
      i. Order-update $\boldsymbol{t}_{i,j}(k)$.
      ii. Evaluate $\mathcal{E}_{i,j}(k)$.
   (b) Set $o_i=\arg\min_{j\in U_i(k)}\mathcal{E}_{i,j}(k)$, $\mathcal{E}_i(k)=\mathcal{E}_{i,o_i}(k)$, $\boldsymbol{t}_i(k)=\boldsymbol{t}_{i,o_i}(k)$ and $U_{i+1}(k)=U_i(k)-\{o_i\}$.

An efficient way of updating (9), instead of a straightforward computation, was proposed in [3] through time- and order-update operations. Among the optimum filters $\boldsymbol{w}_{i,j}(k)$, the one that gives the lowest squared error is used at the current stage and the corresponding user is selected to be the next detected user.

An algorithm exhibiting the same BER performance as the method in [3] but with reduced computationally complexity and increased numerical robustness was proposed in [18], based on the updating of the inverse Cholesky factor of $\boldsymbol{\Phi}_i(k)$. Specifically, if $\boldsymbol{R}_i(k)$ is the upper triangular Cholesky factor of $\boldsymbol{\Phi}_i(k)$, i.e., $\boldsymbol{\Phi}_i(k)=\boldsymbol{R}_i^H(k)\boldsymbol{R}_i(k)$, then the LS solution given in (9) can be expressed as

$$\boldsymbol{w}_{i,j}(k)=\boldsymbol{R}_i^{-1}(k)\boldsymbol{t}_{i,j}(k), \tag{12}$$

where $\boldsymbol{t}_{i,j}(k)$ is defined as

$$\boldsymbol{t}_{i,j}(k)=\boldsymbol{R}_i^{-H}(k)\boldsymbol{z}_{i,j}(k). \tag{13}$$

By substituting (9) in (8) and after some math, we obtain the following expression of the minimum LS error energy for the ith stage and the jth user:

$$\mathcal{E}_{i,j}(k)=\sum_{l=1}^{k}\lambda^{k-l}|d_j(l)|^2-\boldsymbol{w}_{i,j}^H(k)\boldsymbol{z}_{i,j}(k). \tag{14}$$

Finally, using (12)–(14) is rewritten as follows

$$\mathcal{E}_{i,j}(k)=\sum_{l=1}^{k}\lambda^{k-l}|d_j(l)|^2-\|\boldsymbol{t}_{i,j}(k)\|^2. \tag{15}$$

In order to compute the minimum LS error energies $\mathcal{E}_{i,j}$ from (15), the vector $\boldsymbol{t}_{i,j}$ must be computed first. It was shown in [18] that $\boldsymbol{t}_{i,j}(k)$ and $\boldsymbol{R}_i(k)$ can be order updated very efficiently, resulting in significant computational savings.

An extension of the above method to include frequency selective channels was developed in [10], where expanded input and weight vectors are used in order to eliminate both MAI and ISI. Specifically, the input vector of stage i contains the last K_f received vectors, decisions of all users from the past K_b time instances and data decisions from the $i-1$ previous stages at the present time instant. The corresponding equalizer filter is composed of a $PK_f\times 1$ feedforward filter and a $(KK_b+i-1)\times 1$ feedback filter, i.e., with a total of PK_f+KK_b+i-1 taps. Hence, the input of the feedforward filter is described by the vector

$$\underline{\boldsymbol{x}}(k)=[\boldsymbol{x}^T(k-K_f+1)\ \ \boldsymbol{x}^T(k-K_f+2)\ \ \cdots\ \ \boldsymbol{x}^T(k)]^T \tag{16}$$

and the input of the feedback filter is

$$\underline{\boldsymbol{d}}_i(k)=[\boldsymbol{d}^T(k-K_b)\ \ \cdots\ \ \boldsymbol{d}^T(k-1)\ \ d_{o_1}(k)\ \ \cdots\ \ d_{o_{i-1}}(k)]^T, \tag{17}$$

where

$$\boldsymbol{d}(k)=[d_1(k)\ \ d_2(k)\ \ \cdots\ \ d_K(k)]^T \tag{18}$$

is a $K\times 1$ vector containing the decisions for all K users at time k. Note that the decisions $d_i(k)$ correspond to the input symbols at time $k-D$, where D is the equalizer's delay.[5] The $(PK_f+KK_b+i-1)\times 1$ input vector of the ith stage DFE can thus be written as

$$\boldsymbol{y}_i(k)=[\underline{\boldsymbol{x}}^T(k)\ \ \underline{\boldsymbol{d}}_i^T(k)]^T,\quad i=1,2,\ldots,K. \tag{19}$$

Viewing a frequency selective DS-CDMA system as a MIMO system, as presented in (1), the efficient square root LS algorithm of [10] can be straightforwardly applied for multiuser data detection. The resulting scheme will henceforth be referred to as the *square root multiuser detection (SR-MUD)* algorithm. Its steps are summarized in Table 1.

4.2. The RAKE-RLS algorithm

It is important to notice that, in the course of the SR-MUD algorithm, knowledge of users' code sequences is not required. In this section, we will develop an improved version of the SR-MUD algorithm, through incorporating knowledge of the code sequences. This will be done by exploiting the RAKE receiver concept presented in Section 3. We are using here the MIMO formulation of (4), where a DS-CDMA system is represented as a MIMO system with K inputs and $P+L-1$ outputs.

The structure of the new adaptive scheme is shown in Fig. 4. The receiver again detects the users in an ordered manner through K successive stages. At each stage, only one user is detected and the remaining users comprise the

[5] The delays are taken to be equal at all stages, as in [10].

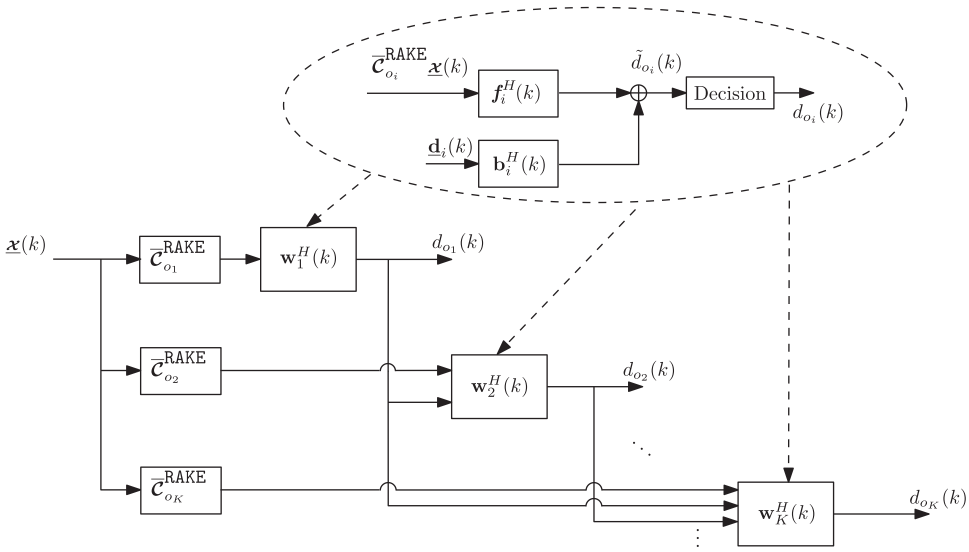

Fig. 4. Structure of the new adaptive DFE equalizer incorporating the knowledge of the code sequences.

set of candidate users for the next stage. A DFE is applied at each stage in order to remove MAI and ISI.

The architecture of the new scheme in Fig. 4 is similar to that of the SR-MUD algorithm, but a modified input signal is applied to the feedforward filter, utilizing the RAKE receiver idea. The equalizer filter $\boldsymbol{w}_i(k)$ of the ith stage DFE is composed of the $K_\mathrm{f}L$-dimensional feedforward filter $\boldsymbol{f}_i(k)$ and the $(KK_\mathrm{b}+i-1)$-dimensional feedback filter $\boldsymbol{t}_i(k)$, as in

$$\boldsymbol{w}_i(k)=[\boldsymbol{f}_i^T(k)\ \ \boldsymbol{b}_i^T(k)]^T,\quad i=1,2,\ldots,K. \tag{20}$$

Given that the o_ith user is to be detected at stage i, the input of the feedforward filter consists of the last K_f received vectors $\overline{\boldsymbol{x}}$ (similarly with the SR-MUD algorithm) but transformed as $\overline{\mathcal{C}}^{\mathrm{RAKE}}_{o_i}\underline{\boldsymbol{x}}(k)$, where

$$\underline{\boldsymbol{x}}(k)=[\overline{\boldsymbol{x}}^T(k-K_\mathrm{f}+1)\ \ \overline{\boldsymbol{x}}^T(k-K_\mathrm{f}+2)\ \ \cdots\ \ \overline{\boldsymbol{x}}^T(k)]^T \tag{21}$$

and $\overline{\mathcal{C}}^{\mathrm{RAKE}}_{o_i}$ is the $K_\mathrm{f}L\times K_\mathrm{f}(P+L-1)$ block diagonal matrix

$$\overline{\mathcal{C}}^{\mathrm{RAKE}}_{o_i}=\boldsymbol{I}_{K_\mathrm{f}}\otimes\mathcal{C}^{\mathrm{RAKE}}_{o_i}=\begin{bmatrix}\mathcal{C}^{\mathrm{RAKE}}_{o_i} & \cdots & \mathbf{0}_{L\times(P+L-1)}\\ \vdots & \ddots & \vdots\\ \mathbf{0}_{L\times(P+L-1)} & \cdots & \mathcal{C}^{\mathrm{RAKE}}_{o_i}\end{bmatrix} \tag{22}$$

with $\mathcal{C}^{\mathrm{RAKE}}_{o_i}$ as in (5). Hence, we have taken advantage of the known code sequences to lessen the effect of the other users. The input of the feedback filter at stage i is the $(KK_\mathrm{b}+i-1)\times 1$ vector $\underline{\boldsymbol{d}}_i(k)$, which contains the decisions for all users from K_b previous time instances and the current data decisions from the $i-1$ previous stages. If the output of the ith stage equalizer, assigned to user o_i, is $\tilde{d}_{o_i}(k)$ and $d_{o_i}(k)=f[\tilde{d}_{o_i}(k)]$ is the corresponding decision device output, then the input of the feedback filter is written as in (17), i.e.,

$$\underline{\boldsymbol{d}}_i(k)=[\boldsymbol{d}^T(k-K_\mathrm{b})\ \ \cdots\ \ \boldsymbol{d}^T(k-1)\ \ d_{o_1}(k)\ \ \cdots\ \ d_{o_{i-1}}(k)]^T, \tag{23}$$

where $\boldsymbol{d}(k)$ is as in (18). Again a decision corresponds to the input symbol of the associated user transmitted D symbol periods earlier.

Using the above definitions, the output of the ith DFE can be written as

$$\begin{aligned}\tilde{d}_{o_i}(k)&=\boldsymbol{f}_i^H(k)\overline{\mathcal{C}}^{\mathrm{RAKE}}_{o_i}\underline{\boldsymbol{x}}(k)+\boldsymbol{b}_i^H(k)\underline{\boldsymbol{d}}_i(k)\\ &=\boldsymbol{w}_i^H(k)\boldsymbol{C}_{i,o_i}\boldsymbol{y}_i(k),\end{aligned} \tag{24}$$

where

$$\boldsymbol{C}_{i,o_i}=\begin{bmatrix}\overline{\mathcal{C}}^{\mathrm{RAKE}}_{o_i} & \mathbf{0}_{K_\mathrm{f}L\times(KK_\mathrm{b}+i-1)}\\ \mathbf{0}_{(KK_\mathrm{b}+i-1)\times K_\mathrm{f}(P+L-1)} & \boldsymbol{I}_{KK_\mathrm{b}+i-1}\end{bmatrix} \tag{25}$$

and

$$\boldsymbol{y}_i(k)=[\underline{\boldsymbol{x}}^T(k)\ \ \underline{\boldsymbol{d}}_i^T(k)]^T,\quad i=1,2,\ldots,K \tag{26}$$

is the $(V+i-1)\times 1$ input vector of the ith stage DFE, where $V=K_\mathrm{f}(P+L-1)+KK_\mathrm{b}$.

Although in the above expressions we have assumed that the detection order is known, practically, we have to specify how it is determined since in time-varying environments the equalizer filters as well as the detection order need to be updated at each stage. In the proposed scheme, the minimization of a LS cost function is used to meet both requirements. To alleviate the error propagation problem, we adopt a BLAST-like methodology where the most reliable signal, in terms of output SNR, is extracted first. The weaker users can be detected with improved performance when stronger users are detected at earlier stages. If the equalizer of the ith stage is to be computed, we have to update all equalizers corresponding to the set of candidate users $U_i(k)$ in order to determine which user should be extracted. The equalizer $\boldsymbol{w}_{i,j}^H(k)$, corresponding to the ith stage and the jth user, is the one minimizing the following cost function:

$$\mathcal{E}_{i,j}(k)=\sum_{l=1}^{k}\lambda^{k-l}\left|d_j(l)-\boldsymbol{w}_{i,j}^H(k)\boldsymbol{C}_{i,j}\boldsymbol{y}_i(l)\right|^2, \tag{27}$$

where $0<\lambda\leq 1$ is the forgetting factor and the matrix $\boldsymbol{C}_{i,j}$ has the structure of $\boldsymbol{C}_{i,o_i}$ in (25), that is, for the ith stage and the jth user, it is given by

$$\boldsymbol{C}_{i,j}=\begin{bmatrix}\overline{\mathcal{C}}_j^{\text{RAKE}} & \boldsymbol{0}_{K_fL\times(KK_b+i-1)}\\ \boldsymbol{0}_{(KK_b+i-1)\times K_f(P+L-1)} & \boldsymbol{I}_{KK_b+i-1}\end{bmatrix}=\begin{bmatrix}\boldsymbol{C}_{i-1,j} & \boldsymbol{0}_{(K_fL+KK_b+i-2)\times 1}\\ \boldsymbol{0}_{1\times(K_f(P+L-1)+KK_b+i-2)} & 1\end{bmatrix}. \tag{28}$$

Remark. Note that setting $\boldsymbol{C}_{i,j}$ equals to

$$\boldsymbol{C}_{i,j}=\begin{bmatrix}\overline{\boldsymbol{\Theta}} & \boldsymbol{0}_{K_fP\times(KK_b+i-1)}\\ \boldsymbol{0}_{(KK_b+i-1)\times K_f(P+L-1)} & \boldsymbol{I}_{KK_b+i-1}\end{bmatrix}, \tag{29}$$

where

$$\overline{\boldsymbol{\Theta}}=\boldsymbol{I}_{K_f}\otimes\boldsymbol{\Theta} \tag{30}$$

with

$$\boldsymbol{\Theta}=[\boldsymbol{I}_P\ \ \boldsymbol{0}_{P\times(L-1)}], \tag{31}$$

the proposed scheme is transformed to the one in Section 4.1, except, of course, for the Cholesky parameterization.

After having updated all tentative equalizers, $\boldsymbol{w}_{i,j}(k)$ for $j\in U_i(k)$, the one achieving the lowest squared error is finally applied at the current stage. In other words, we set

$$o_i=\arg\min_{j\in U_i(k)}\mathcal{E}_{i,j}(k),$$

$$\boldsymbol{w}_i(k)=\boldsymbol{w}_{i,o_i}(k),$$

$$\mathcal{E}_i(k)=\mathcal{E}_{i,o_i}(k). \tag{32}$$

The minimization of (27) with respect to $\boldsymbol{w}_{i,j}(k)$ results in

$$\boldsymbol{w}_{i,j}(k)=[\boldsymbol{C}_{i,j}\boldsymbol{\Phi}_i(k)\boldsymbol{C}_{i,j}^H]^{-1}\boldsymbol{C}_{i,j}\boldsymbol{z}_{i,j}(k)=\boldsymbol{\Phi}_{i,j}^{-1}(k)\boldsymbol{C}_{i,j}\boldsymbol{z}_{i,j}(k), \tag{33}$$

where $\boldsymbol{\Phi}_i(k)$ is the $(V+i-1)\times(V+i-1)$ exponentially time-averaged input autocorrelation matrix, and $\boldsymbol{z}_{i,j}(k)$ the $(V+i-1)\times 1$ crosscorrelation vector, defined as

$$\boldsymbol{\Phi}_i(k)=\sum_{l=1}^{k}\lambda^{k-l}\boldsymbol{y}_i(l)\boldsymbol{y}_i^H(l)=\lambda\boldsymbol{\Phi}_i(k-1)+\boldsymbol{y}_i(k)\boldsymbol{y}_i^H(k), \tag{34}$$

$$\boldsymbol{z}_{i,j}(k)=\sum_{l=1}^{k}\lambda^{k-l}\boldsymbol{y}_i(l)d_j^*(l)=\lambda\boldsymbol{z}_{i,j}(k-1)+\boldsymbol{y}_i(k)d_j^*(k), \tag{35}$$

and

$$\begin{aligned}\boldsymbol{\Phi}_{i,j}(k)&=\boldsymbol{C}_{i,j}\boldsymbol{\Phi}_i(k)\boldsymbol{C}_{i,j}^H\\&=\lambda\boldsymbol{C}_{i,j}\boldsymbol{\Phi}_i(k-1)\boldsymbol{C}_{i,j}^H+\boldsymbol{C}_{i,j}\boldsymbol{y}_i(k)\boldsymbol{y}_i^H(k)\boldsymbol{C}_{i,j}^H\\&=\lambda\boldsymbol{\Phi}_{i,j}(k-1)+\boldsymbol{C}_{i,j}\boldsymbol{y}_i(k)\boldsymbol{y}_i^H(k)\boldsymbol{C}_{i,j}^H.\end{aligned} \tag{36}$$

As it can be seen from (33) and (35), to compute the tentative equalizers $\boldsymbol{w}_{i,j}$ at stage i, current decisions from all users must be known. In the training mode, these are provided by the training sequence. To overcome this causality problem in the decision-directed mode, we assume, as in [3], that the decisions at time k are extracted using the optimum equalizers and detection ordering found at time $k-1$. Hence,

$$\tilde{d}_{o_i}(k)=\boldsymbol{w}_i^H(k-1)\boldsymbol{C}_{i,o_i}\boldsymbol{y}_i(k), \tag{37}$$

$$d_{o_i}(k)=f[\tilde{d}_{o_i}(k)], \tag{38}$$

where o_i here refers to the detection ordering at time $k-1$.

From (26) one can easily see that the equalizer input vector can be expressed in the following order-recursive manner:

$$\boldsymbol{y}_{i+1}(k)=\begin{bmatrix}\boldsymbol{y}_i(k)\\ d_{o_i}(k)\end{bmatrix}. \tag{39}$$

Using this relation in (34), $\boldsymbol{\Phi}_{i+1}(k)$ can be written as

$$\boldsymbol{\Phi}_{i+1}(k)=\begin{bmatrix}\boldsymbol{\Phi}_i(k) & \boldsymbol{z}_{i,o_i}(k)\\ \boldsymbol{z}_{i,o_i}^H(k) & \alpha_{o_i}(k)\end{bmatrix}, \tag{40}$$

where

$$\alpha_j(k)=\sum_{l=1}^{k}\lambda^{k-l}|d_j(l)|^2. \tag{41}$$

Using (36) and with the aid of (40) and (28), we get

$$\begin{aligned}\boldsymbol{\Phi}_{i+1,j}(k)&=\boldsymbol{C}_{i+1,j}\boldsymbol{\Phi}_{i+1}(k)\boldsymbol{C}_{i+1,j}^H\\&=\begin{bmatrix}\boldsymbol{C}_{i,j}\boldsymbol{\Phi}_i(k)\boldsymbol{C}_{i,j}^H & \boldsymbol{C}_{i,j}\boldsymbol{z}_{i,o_i}(k)\\ \boldsymbol{z}_{i,o_i}^H(k)\boldsymbol{C}_{i,j}^H & \alpha_{o_i}(k)\end{bmatrix}\\&=\begin{bmatrix}\boldsymbol{\Phi}_{i,j}(k) & \boldsymbol{C}_{i,j}\boldsymbol{z}_{i,o_i}(k)\\ \boldsymbol{z}_{i,o_i}^H(k)\boldsymbol{C}_{i,j}^H & \alpha_{o_i}(k)\end{bmatrix}.\end{aligned} \tag{42}$$

Invoking the matrix inversion lemma [7], the inverse of $\boldsymbol{\Phi}_{i+1,j}(k)$ can be written as

$$\boldsymbol{\Phi}_{i+1,j}^{-1}(k)=\begin{bmatrix}\boldsymbol{\Phi}_{i,j}^{-1}(k)+\beta_{i,j}(k)\boldsymbol{\Gamma}_{i,j}(k)\boldsymbol{z}_{i,o_i}(k)\boldsymbol{z}_{i,o_i}^H(k)\boldsymbol{\Gamma}_{i,j}^H(k) & -\beta_{i,j}(k)\boldsymbol{\Gamma}_{i,j}(k)\boldsymbol{z}_{i,o_i}(k)\\ -\beta_{i,j}(k)\boldsymbol{z}_{i,o_i}^H(k)\boldsymbol{\Gamma}_{i,j}^H(k) & \beta_{i,j}(k)\end{bmatrix}, \tag{43}$$

where

$$\boldsymbol{\Gamma}_{i,j}(k)=\boldsymbol{\Phi}_{i,j}^{-1}(k)\boldsymbol{C}_{i,j} \tag{44}$$

and

$$\beta_{i,j}(k)=\frac{1}{\alpha_{o_i}(k)-\boldsymbol{z}_{i,o_i}^H(k)\boldsymbol{C}_{i,j}^H\boldsymbol{\Phi}_{i,j}^{-1}(k)\boldsymbol{C}_{i,j}\boldsymbol{z}_{i,o_i}(k)}=\frac{1}{\alpha_{o_i}(k)-\boldsymbol{z}_{i,o_i}^H(k)\boldsymbol{C}_{i,j}^H\boldsymbol{\Gamma}_{i,j}(k)\boldsymbol{z}_{i,o_i}(k)}. \tag{45}$$

Observing Eq. (44), it is easy to see that we can avoid the direct calculation of the product $\boldsymbol{\Phi}_{i,j}^{-1}(k)\boldsymbol{C}_{i,j}$ by order updating $\boldsymbol{\Gamma}_{i,j}(k)$. More specifically, using (28) and (43) in (44), we get

$$\boldsymbol{\Gamma}_{i+1,j}(k)=\boldsymbol{\Phi}_{i+1,j}^{-1}(k)\boldsymbol{C}_{i+1,j}=\begin{bmatrix}\boldsymbol{\Gamma}_{i,j}(k)+\beta_{i,j}(k)\boldsymbol{\Gamma}_{i,j}(k)\boldsymbol{z}_{i,o_i}(k)\boldsymbol{z}_{i,o_i}^H(k)\boldsymbol{C}_{i,j}^H\boldsymbol{\Gamma}_{i,j}(k) & -\beta_{i,j}(k)\boldsymbol{\Gamma}_{i,j}(k)\boldsymbol{z}_{i,o_i}(k)\\ -\beta_{i,j}(k)\boldsymbol{z}_{i,o_i}^H(k)\boldsymbol{C}_{i,j}^H\boldsymbol{\Gamma}_{i,j}(k) & \beta_{i,j}(k)\end{bmatrix}. \tag{46}$$

Note that, although $\boldsymbol{\Gamma}_{i+1,j}(k)$ can be computed from $\boldsymbol{\Gamma}_{i,j}(k)$ for $i=1,2,\ldots,K$, an analytical calculation of $\boldsymbol{\Gamma}_{1,j}(k)$ is necessary at every time instant. By defining the $(V+K)\times K$

matrix

$$\mathbf{Q}(k) = \sum_{l=1}^{k} \lambda^{k-l} [\underline{\mathbf{x}}^T(l)\ \mathbf{d}^T(l-K_b)\ \cdots\ \mathbf{d}^T(l-1)\ \mathbf{d}^T(l)]^T \mathbf{d}^H(l)$$
$$= \lambda \mathbf{Q}(k-1) + [\underline{\mathbf{x}}^T(k)\ \mathbf{d}^T(k-K_b)\ \cdots\ \mathbf{d}^T(k-1)\ \mathbf{d}^T(k)]^T \mathbf{d}^H(k), \tag{47}$$

one can easily find from (35) and (47) that

$$\mathbf{z}_{i,j}(k) = [q_{1,j}(k)\ \cdots\ q_{V,j}(k)\ q_{V+o_1,j}(k)\ \cdots\ q_{V+o_{i-1},j}(k)]^T \tag{48}$$

and

$$\mathbf{z}_{i+1,j}(k) = [\mathbf{z}_{i,j}^T(k)\ q_{V+o_i,j}(k)]^T, \tag{49}$$

where $q_{m,n}(k)$ is the (m,n) entry of $\mathbf{Q}(k)$. Moreover, from (41) and (47), it is straightforward to see that $\alpha_j(k) = q_{V+j,j}(k)$. Finally, an efficient order-recursive relation for the equalizer weights can be obtained by substituting (46) and (49) in (33), as follows:

$$\begin{aligned}
\mathbf{w}_{i+1,j}(k) &= \mathbf{\Phi}_{i+1,j}^{-1}(k)\mathbf{C}_{i+1,j}\mathbf{z}_{i+1,j}(k) \\
&= \mathbf{\Gamma}_{i+1,j}(k)\mathbf{z}_{i+1,j}(k) \\
&= \begin{bmatrix} \mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,j}(k) + \beta_{i,j}(k)\mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,o_i}(k)[\mathbf{z}_{i,o_i}^H(k)\mathbf{C}_{i,j}^H\mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,j}(k) - q_{V+o_i,j}(k)] \\ -\beta_{i,j}(k)\mathbf{z}_{i,o_i}^H(k)\mathbf{C}_{i,j}^H\mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,j}(k) + \beta_{i,j}(k)q_{V+o_i,j}(k) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{w}_{i,j}(k) \\ 0 \end{bmatrix} + \beta_{i,j}(k)[q_{V+o_i,j}(k) - \mathbf{z}_{i,o_i}^H(k)\mathbf{C}_{i,j}^H\mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,j}(k)] \begin{bmatrix} -\mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,o_i}(k) \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{w}_{i,j}(k) \\ 0 \end{bmatrix} + \beta_{i,j}(k)\Delta_{i,j}(k) \begin{bmatrix} -\mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,o_i}(k) \\ 1 \end{bmatrix},
\end{aligned} \tag{50}$$

where

$$\Delta_{i,j}(k) = q_{V+o_i,j}(k) - \mathbf{z}_{i,o_i}^H(k)\mathbf{C}_{i,j}^H\mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,j}(k). \tag{51}$$

In order to determine the optimum detection ordering, the LS error energies must be computed. From (27), (41) and (47), and after some algebra, one can readily arrive at

$$\begin{aligned}
\mathcal{E}_{i,j}(k) &= \sum_{l=1}^{k} \lambda^{k-l} |d_j(l)|^2 - \mathbf{w}_{i,j}^H(k)\mathbf{C}_{i,j}\mathbf{z}_{i,j}(k) \\
&= \alpha_j(k) - \mathbf{w}_{i,j}^H(k)\mathbf{C}_{i,j}\mathbf{z}_{i,j}(k) \\
&= q_{V+j,j}(k) - \mathbf{w}_{i,j}^H(k)\mathbf{C}_{i,j}\mathbf{z}_{i,j}(k).
\end{aligned} \tag{52}$$

An efficient order-recursive formula for the LS error energies $\mathcal{E}_{i+1,j}(k)$ can be derived from (52) using (28), (49)–(51), as follows:

$$\begin{aligned}
\mathcal{E}_{i+1,j}(k) &= \alpha_j(k) - \mathbf{w}_{i+1,j}^H(k)\mathbf{C}_{i+1,j}\mathbf{z}_{i+1,j}(k) \\
&= \alpha_j(k) - \mathbf{w}_{i+1,j}^H(k) \begin{bmatrix} \mathbf{C}_{i,j}\mathbf{z}_{i,j}(k) \\ q_{V+o_i,j}(k) \end{bmatrix} \\
&= \alpha_j(k) - \mathbf{w}_{i,j}^H(k)\mathbf{C}_{i,j}\mathbf{z}_{i,j}(k) \\
&\quad - \beta_{i,j}(k)\Delta_{i,j}(k)[q_{V+o_i,j}(k) - \mathbf{z}_{i,o_i}^H(k)\mathbf{C}_{i,j}^H\mathbf{\Gamma}_{i,j}(k)\mathbf{z}_{i,j}(k)] \\
&= \mathcal{E}_{i,j}(k) - \beta_{i,j}(k)\Delta_{i,j}(k)\Delta_{i,j}(k) \\
&= \mathcal{E}_{i,j}(k) - \beta_{i,j}(k)\Delta_{i,j}^2(k).
\end{aligned} \tag{53}$$

The order of detection is determined if we sort energies in increasing order. The minimum of these energies is denoted by $\mathcal{E}_i(k)$. The user attaining the minimum energy is detected at the next stage.

We have already seen that the equalizer weight vector $\mathbf{w}_{i,j}(k)$ can be order-updated as in (50). However, at every time instant, knowledge of the first-stage filters $\mathbf{w}_{1,j}(k)$, $j=1,2,\ldots,K$, is necessary. This can be accomplished through time-updating $\mathbf{w}_{1,j}(k)$ using the conventional RLS algorithm [7]. Based on (36) and using the matrix inversion lemma, we arrive at the following RLS recursion:

$$\mathbf{p}_j(k) = \mathbf{\Phi}_{1,j}^{-1}(k-1)\mathbf{C}_{1,j}\mathbf{y}_1(k), \tag{54}$$

$$\mathbf{k}_j(k) = \mathbf{\Phi}_{1,j}^{-1}(k)\mathbf{C}_{1,j}\mathbf{y}_1(k) = \frac{\lambda^{-1}\mathbf{p}_j(k)}{1+\lambda^{-1}\mathbf{y}_1^H(k)\mathbf{C}_{1,j}^H\mathbf{p}_j(k)}, \tag{55}$$

$$\mathbf{\Phi}_{1,j}^{-1}(k) = \lambda^{-1}\mathbf{\Phi}_{1,j}^{-1}(k-1) - \lambda^{-1}\mathbf{k}_j(k)\mathbf{p}_j^H(k), \tag{56}$$

$$\mathbf{w}_{1,j}(k) = \mathbf{w}_{1,j}(k-1) + \mathbf{k}_j(k)e_j^*(k), \tag{57}$$

where

$$e_j(k) = d_j(k) - \mathbf{w}_{1,j}^H(k-1)\mathbf{C}_{1,j}\mathbf{y}_1(k) \tag{58}$$

is the *a priori* estimation error. The algorithm presented above is summarized in Table 2. Through time- and order-recursive updates, we efficiently calculate $\mathbf{w}_{i,j}(k)$ and determine the detection ordering o_i. The proposed algorithm will henceforth be referred to as *RAKE-RLS*.

Remark. When the optimum ordering of the users does not change at every symbol instant (e.g., in low mobility scenarios), the update of that ordering can be performed only once every γ symbols, just as in [3,10]. Moreover, the case of some of the users having fixed rank in the user ordering can be taken advantage of in the proposed schemes by simply excluding those user indices from the set of candidate users, $U_i(k)$. Such simplifications can greatly reduce the computational burden of the equalizers, as suggested in the following section for the fixed ordering case.

5. Computational complexity

The computational requirements of the proposed algorithms are assessed in this section. For the sake of simplifying the formulae, let us define the following quantities:

$$T = K_f P + K K_b,$$

$$S = K_f L + K K_b,$$

and recall that $V = (P+L-1)K_f + KK_b$. Then the computational complexities of the proposed algorithms, given in terms of the number of complex multiplications required per symbol time are as follows:

- According to Table 1 of [10], the computational complexity of SR-MUD is as follows:
  $$C_1 = \frac{5}{2}T^2 + \frac{1}{2}TK^2 + \frac{9}{2}TK.$$
- The number of complex multiplications required at each step of the RAKE-RLS algorithm is shown in Table 2. It is easily verified that the overall complexity

Table 2
The RAKE-RLS algorithm. The computational requirements (numbers of complex multiplications) of each step are shown.

Initialization: $k=1$. For $i=1,\ldots,K$, $o_i=i$, $\boldsymbol{w}_i(0)=\mathbf{0}$. For $j=1,\ldots,K$, $\boldsymbol{w}_{1,j}(0)=\mathbf{0}$, $\mathbf{Q}(0)=\mathbf{0}$, $\Phi_{1,j}^{-1}(0)=\delta^{-1}\boldsymbol{I}$, where δ is a small positive constant.

1. $U_1(k)=\{1,2,\ldots,K\}$. For $i=1,2,\ldots,K$
   (a) Calculate $d_{o_i}(k)$ from (37) and (38).
   $\underline{[K(K_f LP+S+\frac{K+1}{2})]}$
2. Time-update matrix $\mathbf{Q}(k)$ using (47). $\underline{[\frac{3}{2}K(V+K)]}$
3. For $j=1,2,\ldots,K$
   (a) Compute $\boldsymbol{p}_j(k)$, $\boldsymbol{k}_j(k)$ and $\boldsymbol{\Phi}_{1,j}^{-1}(k)$ from (54), (55), and (56), respectively. $\underline{[K(K_f LP+\frac{5}{2}S^2+\frac{3}{2}S)]}$
   (b) Time-update $\boldsymbol{w}_{1,j}(k)$ from (57). $\underline{[2KS]}$
   (c) Compute $\boldsymbol{z}_{1,j}(k)$ from (48).
   (d) Evaluate $\mathcal{E}_{1,j}(k)$ from (52). $\underline{[K(K_f LP+S)]}$
   (e) Compute $\boldsymbol{\Gamma}_{1,j}(k)$ from (44). $\underline{[KSK_f LP]}$
4. Set $o_1=\arg\min_{j\in U_1(k)}\mathcal{E}_{1,j}(k)$, $\mathcal{E}_1(k)=\mathcal{E}_{1,o_1}(k)$, $\boldsymbol{w}_1(k)=\boldsymbol{w}_{1,o_1}(k)$ and $U_2(k)=U_1(k)-\{o_1\}$.
5. For $i=2,3,\ldots,K$
   (a) For $j\in U_i(k)$
      i. Compute $\beta_{i-1,j}(k)$, $\Delta_{i-1,j}(k)$ from (45) and (51), respectively. $\underline{[\frac{K(K-1)}{2}(K_f LP+2SV+2S)]}$
      ii. Order-update $\boldsymbol{w}_{i,j}(k)$ from (50). $\underline{[\frac{K(K-1)}{2}S]}$
      iii. Evaluate $\mathcal{E}_{i,j}(k)$ from (53).
      iv. Compute $\boldsymbol{z}_{i,j}(k)$ from (49).
      v. Order-update $\boldsymbol{\Gamma}_{i,j}(k)$ from (46). $\underline{[K(K-1)S(V+1)]}$
   (b) Set $o_i=\arg\min_{j\in U_i(k)}\mathcal{E}_{i,j}(k)$, $\mathcal{E}_i(k)=\mathcal{E}_{i,o_i}(k)$, $\boldsymbol{w}_i(k)=\boldsymbol{w}_{i,o_i}(k)$ and $U_{i+1}(k)=U_i(k)-\{o_i\}$.
6. $k=k+1$. Go to Step 1.

of the algorithm is given by

$$C_2=\frac{K}{2}[1+6S+5S^2+K_f LP(2S+5)+3V-4SV +K(K_f LP+4SV+5S+4)].$$

- The computational complexity of the RAKE-RLS algorithm can be reduced by fixing users' ordering in time. In such a scenario, Step 3 of the algorithm is executed only once. Similarly, Step 5 is performed once for each value of i. The complexity of the algorithm then becomes

$$C_3=\frac{1}{2}\{4K^2+2K_f LP(S+1)+S(5S-8V-1) +K[1+4K_f LP+3V+4S(2V+3)]\}.$$

To have an idea of the relative computational requirements, consider the case of $K=7$ users with spreading gain $P=16$, channel length $L=6$ and DFE lengths $K_f=K_b=1$ as in our simulations. Then $C_1\approx 2611$, $C_2\approx 48\,563$ and $C_3\approx 12\,782$.

It can be seen that RAKE-RLS is significantly more computationally demanding than SR-MUD. The principal reason for this increase in complexity is the incorporation of despreading in this scheme. This operation, although generally leading to an improved performance compared to SR-MUD as shown in the simulations, is responsible for the presence of the $\boldsymbol{C}_{i,j}$ matrices, and hence the dependence of the $\boldsymbol{\Gamma}_i$ matrices on j (not counting the computations needed to multiply with the $\boldsymbol{C}_{i,j}$'s). There is only one such matrix at each stage of SR-MUD, whereas, in RAKE-RLS, such a matrix needs to be considered for each of the candidate users at a given stage (due to the fact that each user employs a different spreading sequence). This fact brings down the computational burden of SR-MUD as compared with RAKE-RLS, in conjunction, of course, with the additional computational savings resulting from its adoption of a Cholesky parameterization [18,10]. Reductions to the complexity of RAKE-RLS as computed above could be achieved via e.g., using fast convolution methods to perform the despreading. Moreover, considerable savings would result if the usual binary (± 1) spreading sequences were used. In that case, the presence of the $\boldsymbol{C}_{i,j}$ matrices would entail no multiplications at all. The number of complex multiplications for RAKE-RLS would then be given by

$$C_4=\frac{K}{2}[1+5S^2+S(6-4V)+3V+K(4SV+5S+4)].$$

In the above example, this amounts to about 35 795 complex multiplications per symbol, a considerable reduction compared to 48 563.

6. Simulations

In this section, the performance of the proposed equalization algorithms is evaluated via computer simulations, in terms of (uncoded) bit error rate (BER). Each user generates QPSK symbols of duration $T_s=0.2\,\mu s$, which are then spread using normalized Walsh–Hadamard code sequences. The generated sequences are transmitted over independent, time-varying and frequency selective channels with Rayleigh-faded paths of an exponential profile. A carrier frequency of 2.4 GHz and a mobile speed of 50 km/h is assumed, yielding a normalized Doppler frequency of about $f_D T_s=2.2\times 10^{-5}$. The feedforward and feedback filters have a temporal span of $K_f=1$ and $K_b=1$ taps, respectively.[6] The equalization delay was chosen for all DFEs as in [10], namely $D=K_f-1=0$.

[6] This is quite reasonable, since, for $L\le P$, at most one previous symbol appears at each received vector in (1) and (4).

The performance of the proposed schemes is compared with that of the RAKE receiver, the ASIC algorithm [12], and the linear receiver (of length P) adapted with the exponentially weighted conventional RLS. The single user bound (SUB), representing the performance of a single-user system in a non-fading AWGN channel, is also shown as a benchmark. For the RAKE receiver, an LMS algorithm [7] is employed to track the channel variations. ASIC also employs LMS, as in [12]. The learning rates (step size for LMS and λ for RLS) were experimentally optimized in each case. A near–far scenario is considered, where the received amplitude of each user is determined such that $10\log_{10}(A_i/A_{i+1})^2 = N$ dB, and the amplitude of the first user is set to 1. In all simulations, frames of 2048 symbols each, with the first 256 devoted to training, were transmitted to obtain the results.

The BER performance versus E_b/N_0 (dB) is depicted in Fig. 5 for $K=7$, $L=6$, $N=2$ dB, and for different values of spreading factor P. As one can see, the superiority of the proposed schemes is evident, especially in the higher E_b/N_0 regime. Specifically, for small values of P ($P=8,16$) and at high E_b/N_0, both schemes achieve better performance compared to the other algorithms, while SR-MUD outperforms RAKE-RLS. The latter is due to the fact that these values of P are relatively small with respect to the channel spread and the number of users. In such scenarios, the RAKE receiver, which underlies RAKE-RLS, is known to have a poor performance [14,1]. Here, this causes RAKE-RLS to lose the advantage it has over SR-MUD coming from the despreading operation. The RAKE receiver, and consequently RAKE-RLS, attains better performance for sufficiently high spreading gain (e.g., [1]). This can be verified from Fig. 5(c) and (d), where one can see that, for relatively large values of P ($P=64$, 128), RAKE-RLS attains the best performance, while the performance of SR-MUD and conventional RLS worsens significantly. Although the performance of the RAKE receiver and the ASIC gets better for larger spreading factors, these are always inferior to RAKE-RLS and SR-MUD in the higher E_b/N_0 regime. The validity of the above observations is also verified in Fig. 6, where the impact of the spreading factor on the algorithms' BER performance is demonstrated for $K=5$, $L=5$, $N=2$ dB and $E_b/N_0=20$ dB.

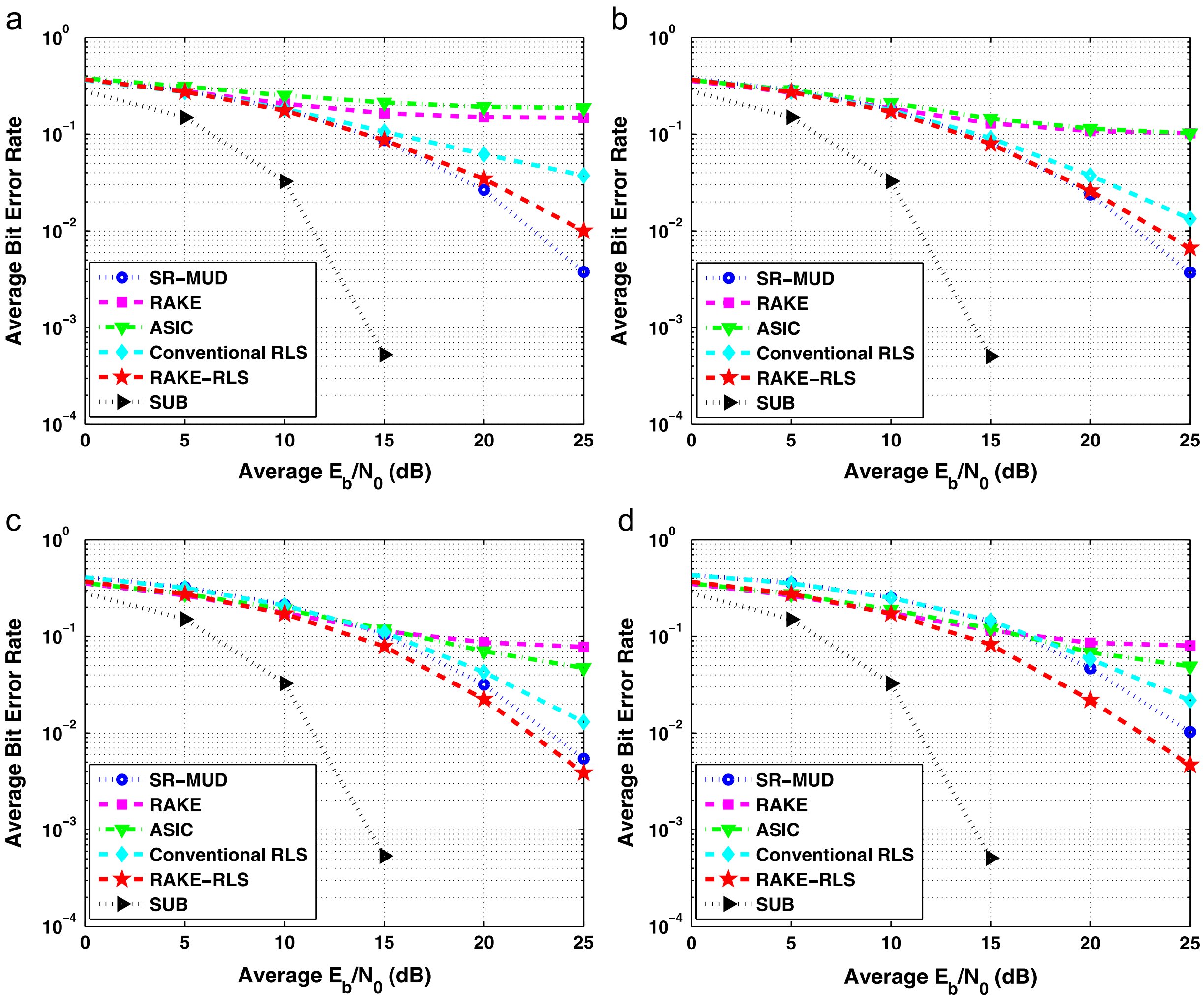

Fig. 5. BER versus E_b/N_0 (dB) for $K=7$, $L=6$, $N=2$ dB and spreading factor (a) $P=8$, (b) $P=16$, (c) $P=64$, (d) $P=128$.

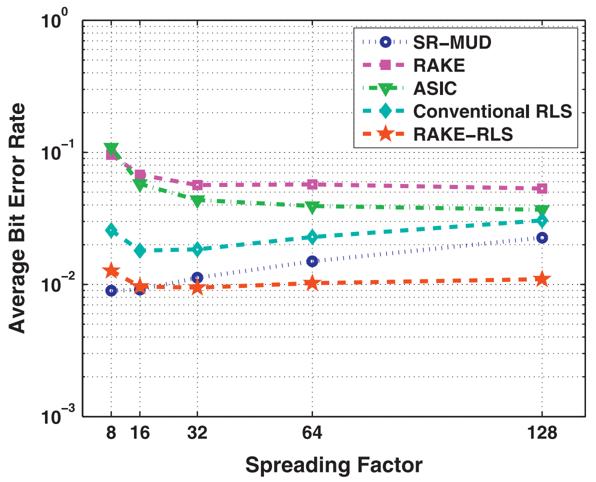

Fig. 6. BER versus spreading factor for $K=5$, $L=5$, $N=2$ dB and $E_b/N_0=20$ dB.

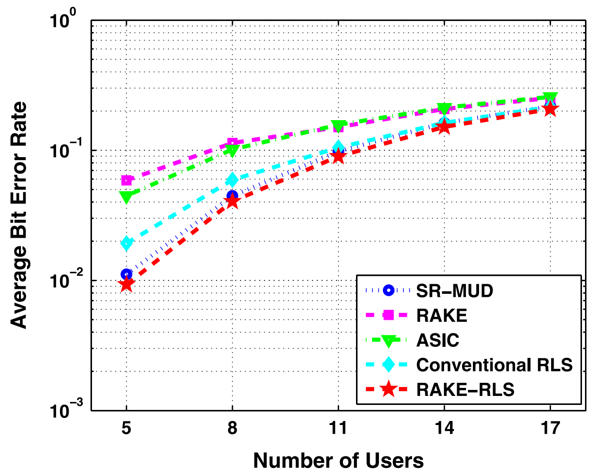

Fig. 8. BER versus number of users for $P=32$, $L=5$, $N=2$ dB and $E_b/N_0=20$ dB.

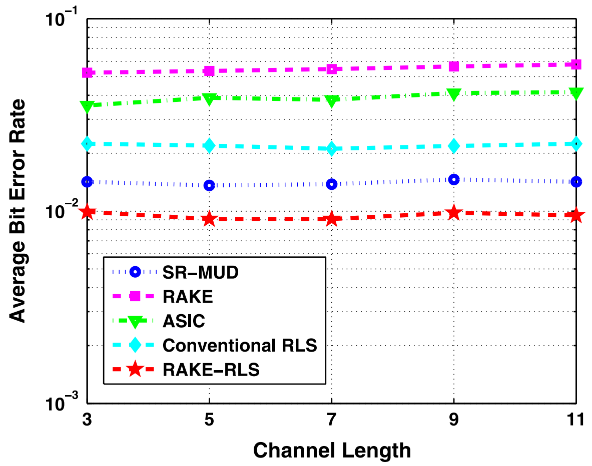

Fig. 7. BER versus channel length for $P=64$, $K=5$, $N=2$ dB and $E_b/N_0=20$ dB.

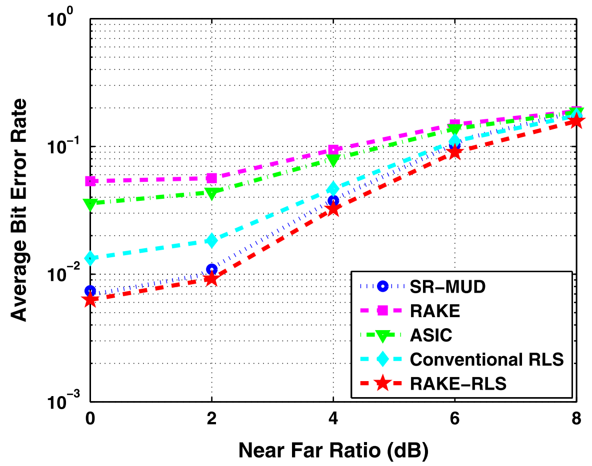

Fig. 9. BER versus near far ratio (dB) for $P=32$, $K=5$, $L=5$ and $E_b/N_0=20$ dB.

Fig. 7 demonstrates the influence of the channel length on the BER performance for $P=64$, $K=5$, $N=2$ dB, and $E_b/N_0=20$ dB. We note that, with relatively large spreading gain, the performance of all algorithms does not change significantly as the channel length increases, while RAKE-RLS achieves the lowest BER. The BER performance versus the number of users is shown in Fig. 8 for $P=32$, $L=5$, $N=2$ dB, and $E_b/N_0=20$ dB. As expected, the BER deteriorates as the number of users increases, while SR-MUD and RAKE-RLS attain the best performance. However, for a large number of users, the performance of all algorithms converges to a high BER. The BER performance versus near far ratio (dB) is depicted in Fig. 9 for $P=32$, $K=5$, $L=5$, and $E_b/N_0=20$ dB. The BER improves as the near–far ratio decreases ($N\rightarrow 0$ dB) and, as expected, the best BER is achieved in the absence of the near–far problem ($N=0$ dB).

The dependence of the performance of the algorithms on the channel fading rate can be seen in Fig. 10, where the BER is plotted against the normalized Doppler spread for a system with $K=7$ users with $N=2$ dB, channels of length $L=6$, and at an SNR of $E_b/N_0=20$ dB, for both small and large spreading factors. Again, and for all speeds considered, the proposed schemes outperform all other algorithms under comparison, while RAKE-RLS is the best for high spreading gains.

The above results suggest that RAKE-RLS can operate efficiently under near–far conditions and in a time-varying, frequency selective environment, especially at medium to high E_b/N_0 values and for relatively larger spreading factors. It keeps the best performance in various scenarios, for increasing channel length, number of users or mobile speed. On the other hand, SR-MUD, which makes no use of the spreading codes, is seen to perform better for relatively smaller values of the spreading factor and at high E_b/N_0 (>20 dB).

Finally, it is worthwhile noting that, as pointed out in Section 4.2, the transition from RAKE-RLS to SR-MUD is straightforward if $\mathbf{C}_{i,j}$ assumes the structure of (29) (modulo, of course, the Cholesky parameterization). Hence, a

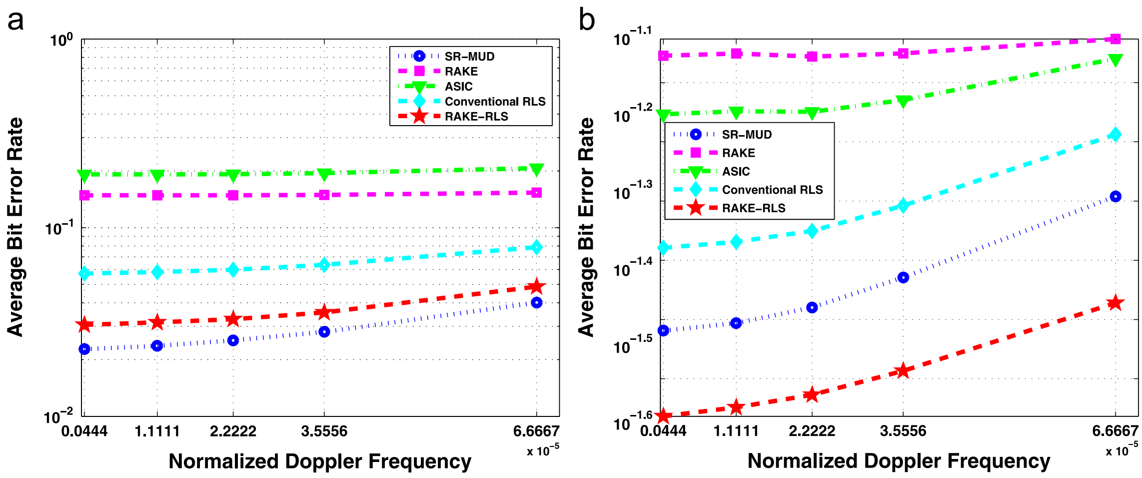

Fig. 10. BER versus normalized Doppler spread for $K=7$, $L=6$, $N=2$ dB, and $E_b/N_0=20$ dB, with (a) $P=8$ and (b) $P=64$.

reconfigurable scheme, which alternates between the two algorithms depending on the value of P employed, could be envisaged to get the best performance among the two, for all spreading factors.

7. Conclusions

Two new adaptive equalization algorithms for time-varying and frequency selective channels in a DS-CDMA system were derived, based on the BLAST idea. The first algorithm results from the application of the idea of [3] to a MIMO-formulated DS-CDMA system, while the second one arises by incorporating the RAKE receiver concept to the first scheme. Both the equalizer filters and the optimum detection ordering are efficiently updated through time- and order-recursions. Improved BER performance is offered compared to existing adaptive DS-CDMA equalizers, in a near–far mobile environment and over a wide range of channel lengths/fading rates and numbers of users.

Acknowledgment

The authors would like to thank Dr. V. Kekatos for providing a Matlab implementation of [10] along with useful explanations. Comments and criticism by the anonymous reviewers are also gratefully acknowledged.

References

[1] H. Boujemaa, M. Siala, On the Rake receiver performance, in: Proceedings of the VTC-2000 (Fall), vol. 3, Boston, MA, 24–28 September 2000, pp. 1483–1488.
[2] Y. Cho, J.H. Lee, Analysis of an adaptive SIC for near–far resistant DS-CDMA, IEEE Transactions on Communications 46 (11) (1998) 1429–1432.
[3] J. Choi, H. Yu, Y.H. Lee, Adaptive MIMO decision feedback equalization for receivers with time-varying channels, IEEE Transactions on Signal Processing 53 (11) (2005) 4295–4303.
[4] G.J. Foschini, G.D. Golden, R.A. Valenzuela, P.W. Wolniansky, Simplified processing for high spectral efficiency wireless communication employing multi-element arrays, IEEE Journal on Selected Areas in Communications 17 (11) (1999) 1841–1852.
[5] G. Ginis, J.M. Cioffi, On the relation between V-BLAST and the GDFE, IEEE Communications Letters 5 (9) (2001) 364–366.
[6] H. Huang, H. Viswanathan, G.J. Foschini, Achieving high data rates in CDMA systems using BLAST techniques, in: Proceedings of the GLOBECOM-1999, vol. 5, Rio de Janeiro, Brazil, December 1999, pp. 2316–2320.
[7] S. Haykin, Adaptive Filter Theory, third ed., Prentice Hall, 1996.
[8] J.M. Holtzman, DS/CDMA successive interference cancellation, in: Proceedings of the ISSSTA-1994, vol. 1, Oulu, Finland, 4–6 July 1994, pp. 69–78.
[9] M. Honig, M.K. Tsatsanis, Adaptive techniques for multiuser CDMA receivers, IEEE Signal Processing Magazine, May, 2000, pp. 49–61.
[10] V. Kekatos, A.A. Rontogiannis, K. Berberidis, Cholesky factorization-based adaptive BLAST DFE for wideband MIMO channels, EURASIP Journal on Advances in Signal Processing 2007 (2007) Article ID 45789.
[11] S. Kun, Z. Xudong, A general multiuser detection method for DS-CDMA system, in: Proceedings of the APCC/MDMC-2004, Tsinghua University, Beijing, China, 29 August–1 September 2004, pp. 448–452.
[12] K.-C. Lai, J.J. Shynk, M. Motamed, R.P. Gooch, Adaptive successive interference cancellation for the IS-95 uplink, in: Proceedings of the VTC-2000 (Fall), vol. 3, Boston, MA, 24–28 September 2000, pp. 1187–1192.
[13] K.-C. Lai, J.J. Shynk, Steady-state analysis of the adaptive successive interference canceler for DS/CDMA signals, IEEE Transactions on Signal Processing 49 (10) (2001) 2345–2362.
[14] M. Latva-aho, Advanced Receivers for Wideband CDMA Systems, Ph.D. Thesis, Department of Electrical Engineering, University of Oulu, Finland, September 1998.
[15] Y.D. Li, T.T. Tjhung, Performance analysis of PIC-CDMA systems, in: Proceedings of the ICCS-2002, vol. 2, Singapore, 25–28 November 2002, pp. 1086–1090.
[16] P. Patel, J. Holtzman, Analysis of a simple successive interference cancellation scheme in a DS-CDMA system, IEEE Journal on Selected Areas in Communications 12 (5) (1994) 796–807.
[17] K. Puttegowda, et al., On the effect of cancellation order in successive interference cancellation for CDMA systems, in: Proceedings of the VTC-2003 (Fall), vol. 2, Orlando, Florida, 6–9 October 2003, pp. 1035–1039.
[18] A.A. Rontogiannis, V. Kekatos, K. Berberidis, A square-root adaptive V-BLAST algorithm for fast time-varying MIMO channels, IEEE Signal Processing Letters 13 (5) (2006) 265–268.
[19] S. Sfar, R.D. Murch, K.B. Letaief, Achieving high capacities in CDMA systems using multiuser detection based on BLAST, in: Proceedings of the ICC-2001, vol. 2, Helsinki, Finland, 11–14 June 2001, pp. 565–569.
[20] S. Verdú, Multiuser Detection, Cambridge University Press, 1998.
[21] G. Wu, et al., Advanced detection scheme for multicode CDMA with V-BLAST architecture, in: Proceedings of the PIMRC-2002, vol. 4, Lisboa, Portugal, 15–18 September 2002, pp. 1810–1814.
[22] W. Zha, S.D. Blostein, Modified decorrelating decision-feedback detection of BLAST space-time system, in: Proceedings of the ICC-2002, vol. 1, New York, NY, 28 April–2 May 2002, pp. 335–339.