MULTIUSER DETECTOR FOR HYBRID CDMA SYSTEMS BASED ON THE BAREISS ALGORITHM

A. Bifano

Dip. Elettronica e Informazione,
Politecnico di Milano,
P.za L. da Vinci 32, I-20133, Milano, Italy.

V. Rampa

I.E.I.I.T. - Sez. Milano
C.N.R., Politecnico di Milano,
P.za L. da Vinci 32, I-20133, Milano, Italy.

ABSTRACT

At the base station in hybrid CDMA systems, multi-user detection (MUD) techniques are exploited to alleviate both multiple access (MAI) and inter-symbol (ISI) interference. MMSE MUD algorithms solve a linear problem where the system matrix has a block-Toeplitz shape. Sub-optimal algorithms with reduced complexity are mandatory to reduce the high computational load imposed by exact inversion techniques. However, when computational complexity is strongly reduced, sub-optimal algorithms may suffer performances degradation and severe near-far effects. In this paper, we introduce a new detector scheme that uses the block-Bareiss algorithm. This detector shows good performances *and* low computational power requirements, comparing favorably with simple implementations based on block-Fourier techniques. Simulations are presented and discussed for the specific TD-SCDMA application.

1. INTRODUCTION

In hybrid CDMA mobile systems, MUD algorithms are adopted at the base station to greatly reduce both inter-symbol (ISI) and multiple access (MAI) interference [1]. For each data-packet, or block, MMSE MUD solves a linear system where the system matrix has a block-Toeplitz structure. Due to the large matrix size and short computational time available, exact MUD computation cannot be employed for the high computational load that it imposes to the base station processing elements. In fact, in real-time systems with limited and expensive computational resources, only sub-optimal algorithms may be efficiently used.

Two main algorithm families are usually employed: block-based MUD [2]-[3]-[4] and one-shot MUD also known as sliding windows detector (SWD) [5]-[6]. Usually, algorithm selection is performed taking into consideration performances, complexity and implementation issues of the algorithms under test. However, when computational power requirement is the key aspect (i.e., during the industrial implementation phase), severe performance degradation may be introduced.

Keeping in mind these considerations, we present a new MUD detector based on the block-Bareiss (BB) algorithm that is derived from the plain Bareiss factorization scheme [7]. For a specific TD-SCDMA application, we compare its performances *and* computational complexity with respect to the corresponding ones of the block-Levinson scheme (BL) [8] and block-Fourier Transform (BFT) algorithm [4]. With respect to the other MUD algorithms, the Bareiss detector is well suited for hardware/software implementation not only for its low computational load but also for its good performances. In addition, it does not suffer from near-far effects that influence very low complexity implementations such as the low order BFT. It is worth noticing that it is also suitable for parallel implementation [9].

The paper is organized as follows: in the next section the hybrid CDMA signal model is described, while Section 3 introduces the Bareiss algorithm and briefly recalls the other reduced complexity algorithms selected for comparison. The performances and computational complexity of the Bareiss method are evaluated in Section 4 and Section 5, respectively. Section 6 draws some conclusions.

2. HYBRID CDMA SIGNAL MODEL

In hybrid CDMA systems, such as in the CWTS standard [10], K users ($1 \leq K \leq 8$) share the same frequency band and time slot while being separated only by different spreading codes. Each k-th user transmits data bursts consisting of $2N$ QPSK symbols ($N = 352/Q$, N symbols for each semi-burst) where the spreading factor Q is assumed here constant for all users ($Q \in \{1,2,4,8,16\}$). In the following, we also impose that the channel impulse response (CIR) is known and do not vary during the burst. Moreover, we assume also that the CIR length W, expressed in chip intervals, is $W = 16$.

The base-band MIMO (Multiple Input Multiple Output) model may be simply expressed at discrete chip time T_c as: $\mathbf{y} = \mathbf{A}\mathbf{d} + \mathbf{n}$. The $M(NQ+W-1) \times 1$ vector $\mathbf{y}$ indicates the signal received at the base station by the array of M antennas ($1 \leq M \leq 8$). Vector $\mathbf{d}$ represents the transmitted data arranged as a vector of size $NK \times 1$ while $\mathbf{n}$ includes the effects of both electronic noise and inter-cell interference. Noise vector $\mathbf{n}$ is zero mean Gaussian, assumed temporally uncorrelated and spatially correlated with covariance matrix $\mathcal{R}_n = E[\mathbf{n}\mathbf{n}^{\mathrm{H}}] = \mathbf{R}_n \otimes I_{NQ+W-1}$ where $\mathbf{R}_n$ is the spatial covariance matrix ($[\mathbf{R}_n]_{m,m} = \sigma_n^2$ for $m = 1,...,M$), $\otimes$ is the Kronecker's product and I_{NQ+W-1} is the unitary matrix of size $NQ+W-1$.

Since CIRs are assumed constant during each data burst, the system matrix $\mathbf{A}$ of size $M(NQ+W-1) \times NK$ can be arranged as N shifted copies of the block $\mathbf{B}$. Fig. 1 shows how each sub-matrix $\mathbf{B}$ of size $M(Q+W-1) \times K$ is composed by K column vectors $\mathbf{b}_k$ where each of them represents the convolution between the k-th spreading code and the CIRs for all M antennas. The term $v = \lceil (Q+W-1)/Q \rceil$ is the delay spread v expressed in symbol intervals. Without any loss of generality, in the followings we will assume a zero-forcing detector scheme (ZF-MUD).

The received signal vector $\mathbf{y}$ is filtered by the whitening space-temporal (S-T) matched filter producing the $NK \times 1$ output vector $\mathbf{y}_{\mathrm{MF}} = \mathbf{A}^{\mathrm{H}}\mathcal{R}_n^{-1}\mathbf{y}$. Linear ZF-MUD is then performed estimating the NK data symbols $\hat{\mathbf{d}} = \mathbf{R}^{-1}\mathbf{y}_{\mathrm{MF}} = \left(\mathbf{A}^{\mathrm{H}}\mathcal{R}_n^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{H}}\ \mathcal{R}_n^{-1}\mathbf{y}$

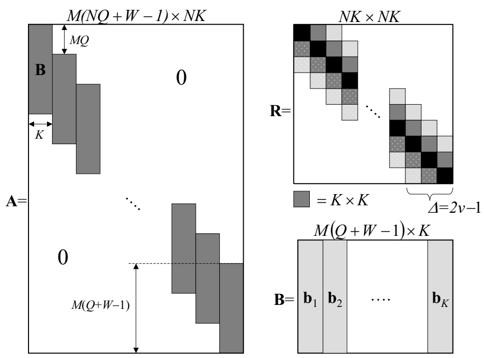

Fig. 1. Matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{R} = \mathbf{A}^{\mathrm{H}}\mathcal{R}_n^{-1}\mathbf{A}$.

where $\mathbf{R}^{-1}$ is the $NK \times NK$ decorrelating matrix obtained from matrix $\mathbf{R} = \mathbf{A}^{\mathrm{H}}\mathcal{R}_n^{-1}\mathbf{A}$.

3. EFFICIENT DETECTOR ALGORITHMS

As shown in Fig. 1, correlation matrix $\mathbf{R}$ is composed by $N \times N$ blocks having size $K \times K$; it is block-Toeplitz and block-band with only $2v-1$ non null block diagonals. Most of the complexity of linear multiuser detector algorithms depends on the inversion methods adopted for this large correlation matrix $\mathbf{R}$ and on the matched filter computation [6]. Using an antenna array with $M >> 1$, the cost of the matched filter $\mathbf{y}_{\mathrm{MF}}$ prevails with respect to the inversion one; however, it is worth noticing that the matched filter calculation has a high degree of parallelism that can be easily exploited using a polyphase Q decimated filter bank.

Some MUD algorithms compute directly the decorrelating matrix $\mathbf{R}^{-1}$and then calculate the solution $\hat{\mathbf{d}} = \mathbf{R}^{-1}\mathbf{y}_{\mathrm{MF}}$ by matrix multiplication. Other algorithms factorize matrix $\mathbf{R}$ and then solve equation $\mathbf{R}\hat{\mathbf{d}} = \mathbf{y}_{\mathrm{MF}}$ by backward/forward substitution (B/FS). While the block-Levinson algorithm belongs to the first class of techniques, the latter family includes the block-Fourier (BFT) [11] algorithm and all methods derived from the QR decomposition. To the second family belongs also the block-Bareiss algorithm that we will introduce later.

3.1. Factorization algorithms

The computation of the Cholesky factor $\mathbf{L}$ for a generic $NK \times NK$ matrix requires a number of operations in the order of $O[N^3K^3]$ which is prohibitive for large block size N and/or high number of users K.

The BFT algorithm [4] is derived from the plain Fourier algorithm: it factorizes matrix $\mathbf{R}$ by means of the Fourier Transform if $\mathbf{R}$ is circulant. Unfortunately, matrix $\mathbf{R}$ is not circulant and it is also block-Toeplitz and block-band. However, it may be made circulant just adding and arranging columns of block. Circulant matrix $\mathbf{R}_c$, that approximates true matrix $\mathbf{R}$, has size $DK \times DK$ where $D = N+v-1$. The block-Fourier algorithm factorizes $\mathbf{R}_c$ by applying a Fast Fourier Transform (FFT) of size D. Solution $\hat{\mathbf{d}}$ is computed by applying a backward substitution and an inverse FFT [4]. It is possible to speed up the block-Fourier algorithm reducing the length D of the FFT with respect to its true value $N+v-1$ by exploiting the well-known overlap-and-save technique and using optimized and inexpensive radix-4 FFT operators. It requires

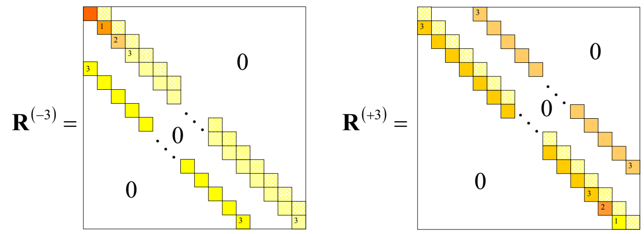

Fig. 2. Structure of matrices $\mathbf{R}^{(-i)}$ and $\mathbf{R}^{(+i)}$ used in the Bareiss algorithm for $i = 3$. The example refers to tridiagonal block matrix $\mathbf{R}$ found in the CWTS standard with $v = 2$.

the use of $L = \lceil N/(D - \mathrm{prelap} - \mathrm{postlap})\rceil$ data vector slices of reduced size D to cover N symbols where prelap, postlap are the overlapping intervals.

The BL algorithm is derived from the plain Levinson algorithm that computes the direct problem $\mathbf{Rx} = \mathbf{y}$ by inverting the matrix $\mathbf{R}$ that is Hermitian, Toeplitz and positive defined. For a generic matrix $\mathbf{R}$ of size $n \times n$, this technique requires a number of operations in the order of $O[n^2]$. The BL algorithm has been extended for block-Toeplitz matrices [8] of size $NK \times NK$ by solving, through $N-1$ iterative steps, the block-Toeplitz system $\mathbf{R}^{(i+1)}\hat{\mathbf{d}}^{(i+1)} = \mathbf{y}_{\mathrm{MF}}^{(i+1)}$ where $\hat{\mathbf{d}}^{(i)}$and $\mathbf{y}_{\mathrm{MF}}^{(i)}$ are sub-vectors of length iK obtained from vectors $\hat{\mathbf{d}}$ and $\mathbf{y}_{\mathrm{MF}}$ respectively. As in the BFT algorithm, it is also possible to speed up the algorithm processing by considering that after few iterations, some internal parameters converge rapidly to their final values [4] and may therefore considered constant. All these approximate algorithms have different behavior depending on one or more parameters: FFT size D and prelap/postlap (or L) parameters for BFT and iteration step i for BL. These design parameters greatly affect both performances, computational complexity and hardware/software implementation issues.

3.2. Bareiss algorithms

The plain Bareiss algorithm [7] employs an iterative technique for solving a generic linear Toeplitz system $\mathbf{Rx} = \mathbf{y}$ by LU factorization of the system matrix $\mathbf{R}$ of size $n \times n$. The complexity of the algorithm is in the order of $O[n^2]$. The key point is to transform the original system in the following equivalent ones (for $1 \le i \le n-1$):

$$\mathbf{R}^{(\pm i)}\mathbf{d} = \mathbf{y}^{(\pm i)} \tag{1}$$

where the matrices $\mathbf{R}^{(-i)}$ have zero elements along the i sub-diagonals below the main diagonal and matrices $\mathbf{R}^{(+i)}$ have zero elements along the i sub-diagonals above the main diagonal. As shown in Fig. 2, $\mathbf{R}^{(-i)}$ is an upper triangular matrix while $\mathbf{R}^{(+i)}$ is a lower triangular one. These transformations are chosen to eliminate diagonals; each transformation requires a number of operations in the order of $O[n]$.

Let $\mathbf{Z}_i = \left(\delta_{w-j+i}\right)_{w=0,\dots,n-1;\, j=0,\dots,n-1}$ be a shift matrix such that pre-multiplication by $\mathbf{Z}_{+i}$ shifts i rows up with zero fill and pre-multiplication by $\mathbf{Z}_{-i}$ shifts i rows down with zero fill. The transformations of the i-th iteration are defined by:

$$\begin{aligned}
\mathbf{R}^{(-i)} &= \mathbf{R}^{(-(i-1))} - m_{-i}\mathbf{Z}_{-i}\mathbf{R}^{+(i-1)} \\
\mathbf{y}^{(-i)} &= \mathbf{y}^{(-(i-1))} - m_{-i}\mathbf{Z}_{-i}\mathbf{y}^{+(i-1)} \\
\mathbf{R}^{(+i)} &= \mathbf{R}^{(+(i-1))} - m_{+i}\mathbf{Z}_{+i}\mathbf{R}^{-(i)} \\
\mathbf{y}^{(+i)} &= \mathbf{y}^{(+(i-1))} - m_{+i}\mathbf{Z}_{+i}\mathbf{y}^{+(i)}
\end{aligned} \tag{2}$$

for $i = 1,\ldots,n-1$ where $m_{-i} = r_{i,0}^{(-(i-1))}/r_0$, $m_{+i} = r_{+i,0}^{(i-1)}/r_{n-1,n-1}^{(-i)}$ and δ_i is the Kronecker's delta. After $n-1$ iterations, the system $\mathbf{R}^{(-(n-1))}\mathbf{d} = \mathbf{y}^{(-(n-1))}$ may be solved with BS. This method is equivalent to the LU factorization of $\mathbf{R} = \mathbf{LU}$: $r_0\mathbf{L} = \left(\mathbf{R}^{(n-1)}\right)^{T2}$ and $\mathbf{U} = \mathbf{R}^{(-(n-1))}$ where ^{T2} denotes matrix transposition above the main diagonal.

The block-Bareiss algorithm still employs (1) but now transformations try to eliminate block diagonals of size $K \times K$. After i iterations, the matrix $\mathbf{R}^{(-i)}$ has null blocks along the i sub-diagonals below the main diagonal and the matrix $\mathbf{R}^{(+i)}$ has null blocks along the i sub-diagonals above the main diagonal. The transformations of the i-th iteration ($i = 1,\ldots,N-1$) are defined by equations:

$$\begin{aligned}
\mathbf{R}^{(-i)} &= \mathbf{R}^{(-(i-1))} - \mathbf{M}_{-i}\mathbf{Z}_{-i}\mathbf{R}^{(i-1)}\\
\mathbf{y}^{(-i)} &= \mathbf{y}^{(-(i-1))} - \mathbf{M}_{-i}\mathbf{Z}_{-i}\mathbf{y}^{(i-1)}\\
\mathbf{R}^{(+i)} &= \mathbf{R}^{(+(i-1))} - \mathbf{M}_{+i}\mathbf{Z}_{+i}\mathbf{R}^{(-i)}\\
\mathbf{y}^{(+i)} &= \mathbf{y}^{(+(i-1))} - \mathbf{M}_{+i}\mathbf{Z}_{+i}\mathbf{y}^{(-i)},
\end{aligned} \tag{3}$$

where $\mathbf{Z}_i = \left(\delta_{w-j+i}\right)_{w=0,\ldots,N-1;j=0,\ldots,N-1}$, $\mathbf{M}_{-i} = \mathbf{R}_{i,0}^{(-(i-1))}\left(\mathbf{R}_0\right)^{-1}$ and $\mathbf{M}_{+i} = \mathbf{R}_{+i,0}^{(i-1)}\left(\mathbf{R}_{N-1,N-1}^{(-i)}\right)^{-1}$. After $N-1$ iterations, we obtain an upper triangular system $\mathbf{R}^{(-(N-1))}\mathbf{d} = \mathbf{y}^{(-(N-1))}$ where $\mathbf{R}^{(-(N-1))}$ is a block matrix. It may be solved by applying a B/FS. Note that, as in the BL algorithm, the most important design parameter is the number i of iterations.

4. SIMULATION RESULTS

According to the CWTS standard [10], the parameters employed in system simulations are: $K = 8$ ($K = 2$ in near-far simulations), $N = 22$, $Q = 16$ and $W = 16$. The scenario is characterized by 100000 bursts of up-link traffic data only with scrambling and spreading codes but without channel coding. The spatio-temporal channel model adopted for the simulations is derived from the Typical Urban (TU) multipath propagation channel as introduced by the COST-207 group. This model has been selected for easy comparison with respect to spatial only algorithms ($M = 1$). For each k-th user, the channel H_k consists of a single cluster of 12 uncorrelated paths, whose delays and mean powers are fixed and set according the TU model. Each p-th path angle ($p = 1,..,12$) is a random variable $\theta_{k,p} = \mathcal{N}\left(\theta_k, \sigma_\theta^2\right)$ with $\theta_k = \mathcal{U}[-\pi/3, +\pi/3]$ and $\sigma_\theta = \pi/36$. Mobile position is not changed during simulation. In addition, perfect knowledge of the channel is assumed while $E\left[\|H_k\|^2\right]$ is constant for all K users. The base station employs a single antenna ($M = 1$) or a linear array ($M = 8$) of equally spaced antennas at $\lambda/2$. The performances of the approximate MUD algorithms (BFT, BB and BL) and the exact inversion scheme are compared in terms of BER for varying SNR defined as $SNR = QE\left[\|H_k\|^2\right]/2\sigma_n^2$. In the followings, the block-Fourier algorithms are indicated as BFT-4 and BFT-16 for $D = 4$ and $D = 16$, respectively.

Fig. 3 shows algorithm performance degradation due to near-far effects with single antenna array $M = 1$, $K = 2$ users and perfect ($\rho = 0$ dB) or no power control ($\rho = 20$ dB). While BB and BL algorithms show good near-far resistance, both BFT-4 and BFT-16 are not well suited in case of an inefficient power control system.

In Fig. 4 and 5, the performances of the MUD algorithms are compared in the case of a single antenna ($M = 1$) and antenna array ($M = 8$), respectively. In the single antenna scenario, for two

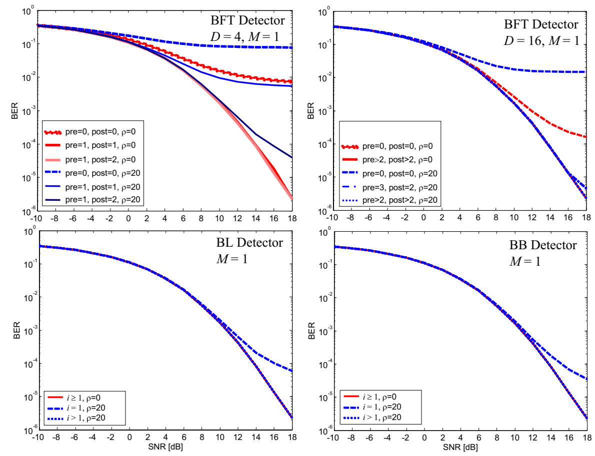

Fig. 3. Near-far performances of MUD algorithms in terms of BER vs. SNR for different design parameters (D, prelap, postlap, i). Simulation parameters are: $\rho = 0, 20$ dB, $M = 1$, $K = 2$, $Q = 16$, $W = 16$.

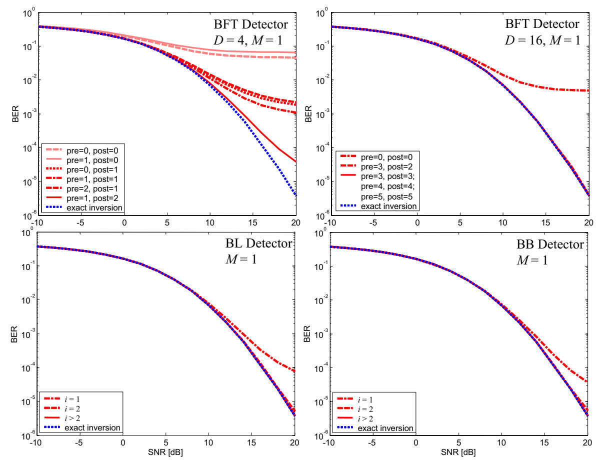

Fig. 4. Performances of MUD algorithms in terms of BER vs. SNR for different design parameters (D, prelap, postlap, i). Simulation parameters are: $M = 1$, $K = 8$, $Q = 16$, $W = 16$.

or more iterations ($i \geq 2$) both BB and BL algorithms have performances similar to the ones corresponding to the exact system inversion method. On the contrary, the BFT algorithm strongly depends on the FFT size D and the prelap (pre) and postlap (post) values. In fact, while for $D = 16$, pre > 0 and post > 0, the BFT detector performances are close the the optimum ones, if $D = 4$ any combination of pre and post coefficients cannot obtain performances near to those corresponding to the exact inversion case, being combination pre $= 1$, post $= 2$ the most effective ($L = 22$).

For the linear antenna array scenario with $M = 8$, performances are significantly better than the previous ones due to the effect of the array spatial processing. All algorithms show good performances except the BFT algorithm for the artifacts introduced by the overlap-and-save technique (e.g., $D = 16$, pre $= 0$, post $= 0$; $D = 4$, pre $= 0$, post $= 0$; $D = 4$, pre $= 1$, post $= 0$ and $D = 4$, pre $= 0$, post $= 1$). It is worth noticing that only one iteration is enough for BB and BL algorithms to converge to the exact solution.

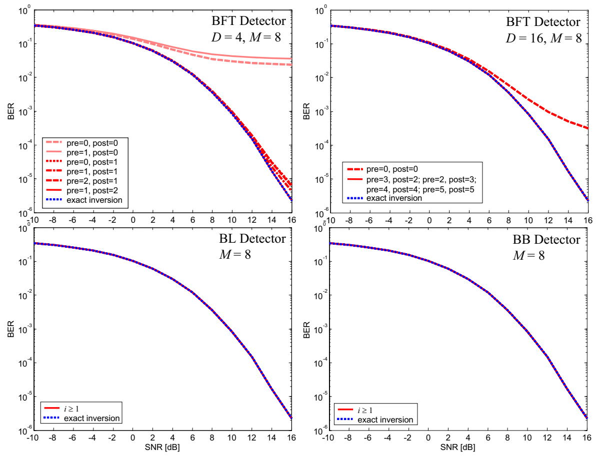

Fig. 5. Performances of MUD algorithms in terms of BER vs. SNR for different design parameters (D, prelap, postlap, i). Simulation parameters are: $M = 8$, $K = 8$, $Q = 16$, $W = 16$.

5. ALGORITHM COMPUTATIONAL COMPLEXITY

Tab. 1 shows the computational complexity of BB, BFT and BL algorithms in terms of complex multiplications ($\times 10^5$) for each semi-burst. The shown algorithms perform the matched filter in the time domain (e.g., BL and BB) or frequency domain (e.g., BFT). Matrix $\mathbf{R}$ is tri-diagonal ($v = 2$) block-Toeplitz, with a lot of null blocks. Several algorithm optimizations have been adopted to reduce both computational complexity and storage size of the algorithms shown in section 3. For instance, in the BB algorithm it is possible to: reduce the matrix multiplications of (3) to block multiplications; rearrange the matrices of blocks $\mathbf{R}^{(+i)}$ and $\mathbf{R}^{(-i)}$ as one column of blocks of variable size for each modified diagonal; reuse the already inverted blocks in the backward substitution phase [12]. It is apparent that both BB and BFT-4 algorithms are well suited for their low complexity requirements. However, when overall performances are the key aspect, the Bareiss algorithm should be employed.

Param.	Algorit.	Comp. complexity	
		$M=1$	$M=8$
$L=4$	BFT-16	0.27	1.19
$L=3$		0.22	1.01
$L=2$		0.18	0.83
$L=22$	BFT-4	0.25	1.19
$L=11$		0.14	0.68
$L=6$		0.09	0.45
$i=21$	BB	1.43	1.97
$i=5$		0.41	0.94
$i=3$		0.28	0.82
$i=1$		0.15	0.69
$i=21$	BL	3.68	4.21
$i=5$		1.06	1.59
$i=3$		0.91	1.45
$i=1$		0.81	1.34

Table 1. Computational complexity of detector algorithms in terms of complex multiplications ($\times 10^5$) per semi-burst.

6. CONCLUSIONS

A novel CWTS multiuser detector based on block-Bareiss algorithm has been introduced. From the computational point of view, it has a complexity figure similar to the block-Fourier algorithm. However, the block-Bareiss shows better near-far resistance and overall performances with respect to the Fourier detector.

7. ACKNOWLEDGMENTS

The authors would like to thank E. Borello, L. Costa, C. Madè and C. Meregalli for the simulation software of the CWTS system. The authors would like also to show appreciation to M. Nicoli for helpful discussions.

8. REFERENCES

[1] S. Verdú, "Multi-user Detection," Cambridge University Press, 1998.

[2] Y. Pigeonnat, "Joint detection for UMTS: complexity and alternative solutions", Proc. IEEE VTC 1999, Vol. 1, pp. 546-550, 1999.

[3] H. R. Karimi, N. W. Anderson, "A novel and efficient solution to block-based joint-detection using approximate Cholesky factorization", Proc. IEEE PIMRC 1998, Vol. 3, pp. 1340-1345, 1998.

[4] M. Vollmer, M. Haardt, J. Götze, "Comparative Study of Joint-Detection Techniques for TD-CDMA based Mobile Radio Systems", Sel. Areas in Communications, Vol. 19, N. 8, pp. 1461-1475, Aug. 2001.

[5] M. J. Juntti, B. Aazhang, "Finite memory length linear multiuser detection for asynchronous CDMA communications", IEEE Trans. on. Comm. Vol. 54, N. 5, pp. 611-622, May 1997.

[6] M. Beretta, A. Colamonico, M. Nicoli, V. Rampa, U. Spagnolini, "Space-Time multiuser detectors for TDD-UTRA: design and optimization", Proc. IEEE VTC 2001, Vol. 1, pp. 78-82, 2001.

[7] E.H. Bareiss, "Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices", Numer. Math. Vol. 13, pp. 404-424, 1969.

[8] R. A. Wiggins, E. A. Robinson, "Recursive solutions to the multichannel filtering problem", J. Geophys. Res., Vol. 70, pp. 1885–1891, 1965.

[9] R. P. Brent, "Parallel algorithms for Toeplitz systems", Numerical Linear Algebra, Digital Signal Processing, Proc. NATO ASI, Leuven, Belgium, August 1988, edited by G. H. Golub and P. Van Dooren, NATO ASI Series F: Computer and Systems Sciences, Vol. 70, Springer-Verlag, 1991

[10] CWTS, "China Wireless Telecommunication Standard (CWTS) Working Group 1 (WG1): Physical channels and mapping of transport channels onto physical channels", TS C102 v3.3.0 (2000-09).

[11] M. Vollmer, M. Haardt, J. Götze, "Shur algorithms for joint-detection in TD-CDMA based mobile radio systems", Ann. Telecommun. Vol. 54, N. 7-8, pp. 365-378, 1999.

[12] E. Borello, L. Costa, "Algoritmi e Architetture efficienti per la rivelazione multi-utente nel sistema CWTS", Thesis as Dottore in Ingegneria (in italian), Dipartimento di Elettronica e Informazione, Politecnico di Milano, 2004.