Space-Time multiuser detectors for TDD-UTRA: design and optimization.

M. Beretta(1), A. Colamonico(2), M. Nicoli(1), V. Rampa(3), U. Spagnolini(1)
(1)D.E.I. - Politecnico di Milano, Piazza L. Da Vinci 32, Milano, (MI), Italy
(2)Siemens ICN, Cinisello Balsamo, (MI), Italy
(3)C.S.T.S. - C.N.R. - Politecnico di Milano, Via Ponzio 34/5, Milano, (MI), Italy
e-mail: {beretta,nicoli,rampa,spagnoli}@elet.polimi.it,
Armando.Colamonico@icn.siemens.it

Abstract— **Linear multiuser detection (MUD) for frequency selective channels has always been considered a prohibitive computational task in CDMA systems. In time slotted CDMA, the block-type MUD involves the inversion of a large matrix that depends on the block size and on the number of users.**

Sub-optimal techniques are computationally efficient but show some performance degradation. The reduced complexity detectors can be either block-type or one-shot. Compared to one-shot approximation of MUD, the block-type detectors have less computational complexity and large latency. However, the tracking of channel variations within the block is not feasible with any block-type processing (e.g., for the adaptive receiver).

These compelling aspects force us to use a one-shot MUD algorithm for space-time channels such as the sliding window decorrelator (SWD). Block-based MUD and SWD algorithms for TDD-UTRA system are compared in term of performance, computational complexity, parallelism and hardware implementation.

I. Introduction

In CDMA systems, the adaptive antenna array is adopted in the up-link to reduce the interference but it requires appropriate space-time (S-T) processing. Linear multi-user detection for time-dispersive channels has always been considered a prohibitive computational task when the block-size and/or the number of users are too large. This becomes even worse when S-T MUD has to be carried out. Sub-optimum techniques that approximate the ideal MUD algorithms are computationally efficient but exhibit some performance degradation.

Efficient computational solutions for block-based receivers for UMTS TDD-UTRA are available [1], [4]. However, the intrinsic latency caused by the block-style processing limits the possibility to update the channel estimation from decisions (adaptive receiver). This fact motivates the use of a S-T MUD for reduced block sizes such as the SWD approach [6]. In addition, the SWD algorithm is well suited to be optimized in parallel hardware architectures.

This paper is focused on the comparison between block-based MUD and SWD algorithms for S-T processing in term of performance, computational complexity and parallelism; further remarks about hardware implementation based on pipelined systems (systolic arrays) are presented.

II. S-T processing model

In time slotted CDMA systems such as in the TDD-UTRA standard (in this section, the corresponding system parameters are indicated within brackets), K users ($K \leq 8$) are active in the same frequency band and in the same time-slot. Each user transmits a burst consisting of two data blocks (semi-bursts) of N QPSK symbols each ($N = 976/Q \leq 976$ for burst type 1) with spreading factor Q ($Q \leq 16$).

The discrete-time model for the uplink, that accounts for both multiuser access (MAI) and intersymbol interference (ISI), can be obtained by sampling at the chip rate the output of the chip matched filter. For the single semi-burst, the signals received at the base station by the array of M antennas, each with temporal channel length W expressed in chip intervals ($W \leq 57$), can be arranged into the $M(NQ+W-1)\times 1$ vector $\mathbf{y} = \mathbf{Ad} + \mathbf{n}$, where $\mathbf{n}$ represents the noise and the intercell interference, $\mathbf{d}$ is the data vector of length NK and $\mathbf{A} = [\mathbf{A}_1^H ... \mathbf{A}_M^H]^T$. Each submatrix $\mathbf{A}_m$ of size $(NQ+W-1)\times KN$ contains shifted copies of the composite signatures $\mathbf{S}_m$ representing the convolution between the spreading codes and propagation channels (see Fig. 1).

Without any loss of generality, we consider that the noise $\mathbf{n}$ is white Gaussian and we adopt the decorrelating (or zero forcing) multi-user detector for the estimation of the data symbols $\hat{\mathbf{d}}$. Moreover, known channel state information is assumed. The algorithms discussed in this paper can be easily generalized to systems with correlated noise and/or MMSE MUD (with or without decision feedback).

The MUD algorithm can be reduced into two steps: at first, the signals $\mathbf{y}$ are filtered by the S-T matched filter to get the $NK \times 1$ vector $\mathbf{y}_{MF} = \mathbf{A}^H\mathbf{y}$. Then linear decorrelation is performed on the matched filter output to estimate the NK data symbols: $\hat{\mathbf{d}} = \mathbf{R}^{-1}\mathbf{y}_{MF}$ where $\mathbf{R} = \mathbf{A}^H\mathbf{A}$.

III. Detector computational complexity

Computation complexity of the linear detector for S-T channels arises from the calculation of the matched filter $\mathbf{y}_{MF}$ and of the matrix inversion $\mathbf{R}^{-1}$. The latter is performed exploiting the Cholesky factor $\mathbf{L}$ of the large block Toeplitz matrix $\mathbf{R} = \mathbf{A}^H\mathbf{A}$. It is worth noticing

0-7803-7005-8/01/$10.00 © 2001 IEEE

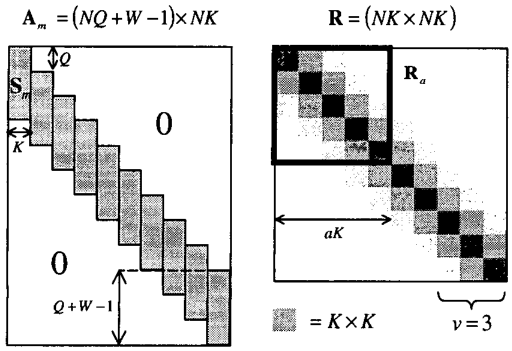

Fig. 1. Matrices $\mathbf{A}_m$ and $\mathbf{R} = \mathbf{A}^H\mathbf{A}$.

that the cost of the matched filter (MF) increases linearly with the number M of antenna; however, it can be moderately optimized just considering that it may be performed in parallel for each antenna. When the Cholesky factor $\mathbf{L}$ is computed, the lower triangular system $\mathbf{L}^H\mathbf{x} = \mathbf{y}_{MF}$ is solved by forward substitution using the intermediate vector $\mathbf{x}$. Finally, the upper triangular system $\mathbf{L}\hat{\mathbf{d}} = \mathbf{x}$ is computed using the backward substitution algorithm. Let $v = \lceil (Q + W - 1)/Q \rceil$ be the delay spread expressed in symbol intervals. The Cholesky factorization requires a number of operations in the order of $O[K^3N^3]$ that is prohibitive for large block size N and when the number of users K increases. The matrix $\mathbf{R}$ is block Toeplitz and block band with only $2v-1$ non null diagonal blocks (Fig. 1), therefore it is possible to introduce efficient algorithms to reduce the computational complexity.

In the following part of this paper the computational complexity of the algorithms considered for MUD is shown for 2 data blocks of N symbols (2 semi-bursts of the TDD-UTRA standard) and expressed in terms of real multiplications (or, equivalently, multiply and accumulation operations - MAC).

A. Exact detector

The Cholesky factorization algorithm for the block band matrix $\mathbf{R}$ can reduce the computational complexity in the order of $O[2NK^3v^2]$ [5]. On the contrary, exploiting the block Schur algorithm for the QR decomposition of a sparse block Toeplitz matrix as shown in reference [4], it is further possible to reduce the complexity in the order of $O[8NK^3v]$. For both BB and BS algorithms, the computational complexity with respect to the number of antennas is linear, being in the order of $O[2MQK^2v^2]$. It is worth noticing that these algorithms are well suited for propagation channels with small temporal spread v.

Table I shows the computational complexity (second column from left) of the block-type MUD algorithms based on the exact Cholesky factorization for block band matrices (BB) and the Schur algorithm for block Toeplitz matrices (BS). For the Cholesky algorithm, the computational complexity takes into account also the computation of the term $\mathbf{R} = \mathbf{A}^H\mathbf{A}$. For the Schur algorithm the derivation of the generators is included ($\mathbf{R}$ is not calculated explicitly, see reference [4]). In addition, Table I specifies also the computational load for $M = 1$ (third column from left) and $M = 8$ (fourth column from left) using parameter values from the TDD-UTRA standard. The computational complexity for the matched filter (MF), the backward/forward substitution (B/F), the Cholesky factorization (using the block banded factorization algorithm and the block Schur algorithm) is shown here.

TABLE I

MUD COMPUTATIONAL COMPLEXITY (IN TERMS OF REAL MULTIPLICATIONS $\times 10^5$) USING BLOCK-TYPE ALGORITHMS FOR THE EXACT CHOLESKY FACTORIZATION (CHOLESKY FACTORIZATION FOR BLOCK BAND MATRICES - BB - AND THE SCHUR ALGORITHM FOR BLOCK TOEPLITZ MATRICES - BS). THE NUMERICAL VALUES ARE CHOOSEN AS $K = 8$, $v = 5$, $N = 61$, $Q = 16$, $W = 57$.

Operation	Comp. complexity		
		$M = 1$	$M = 8$
MF	$O(8QvMKN)$	2.8	22.4
B/F	$O(16vK^2N)$	2.7	2.7
Chol. fact. (BB)	$O(2v^2K^3N)$	13.2	16.8
Chol. fact. (BS)	$O(8vK^3N)$	10.7	14.3
Total MUD (BB)		18.7	41.9
Total MUD (BS)		16.2	39.4

It is apparent that the MF load increases linearly with M; for instance, if $M = 8$, the MF blocks accounts for most of the computational complexity of the detector. However, its calculation may be highly parallelized (e.g., using one MF processor for each antenna). Even the Cholesky factorization is a very demanding task especially when M is low. The exact factorization is not feasible in practical applications and an optimized detector based on an approximate Cholesky factorization, even if sub-optimum, is advisable.

B. Approximate detectors

The Cholesky factor $\mathbf{L}$ of a block multidiagonal Toeplitz matrix $\mathbf{R}$ is not block Toeplitz itself even if it is still block diagonal. However, it was proved in [3] that, if $N \to \infty$ and the number $2v-1$ of the non-zero blocks remains finite, the block rows of the Cholesky factor $\mathbf{L}$ converge to the same block row. Therefore, if N is large and $v << N$, the Cholesky factor $\mathbf{L}$ is nearly block Toeplitz. It is possible to obtain $\mathbf{L}$ just calculating the Cholesky factor of the first a block columns of $\mathbf{R}$ and then copying them in $\mathbf{L}$ for all N columns. This is equivalent to compute the Cholesky factor (e.g., using either the Cholesky factorization algorithm for block band matrices or the Schur algorithm for block-Toeplitz matrices seen before) of the top left $aK \times aK$ sub-matrix $\mathbf{R}_a$ and then copying the smaller blocks (see also Fig. 1). This method will be referred to as approximate Cholesky decorrelator (ACD). The length a must be chosen large enough to avoid edge effect: $a \geq v$.

The sliding window decorrelator (SWD) [6] is an alternative low complexity detector based on the segmentation of the block $\mathbf{y}_{MF}$ into smaller sub-blocks $\mathbf{y}_{MF}^{(i)}$ of size aK (see Fig. 2). For each observation window, only the central symbols of all K users are detected. This reduces the computation to $N-a+1$ linear systems of size $aK \times aK$. Each linear system can be solved by the Cholesky factorization of $\mathbf{R}_a$ followed by the appropriate backward and forward substitutions. The influence due to the neighboring symbols outside the window is negligible if the window size a of the SWD algorithm is larger than the bandwidth of matrix $\mathbf{R}$ (i.e., $a \geq 2v-1$).

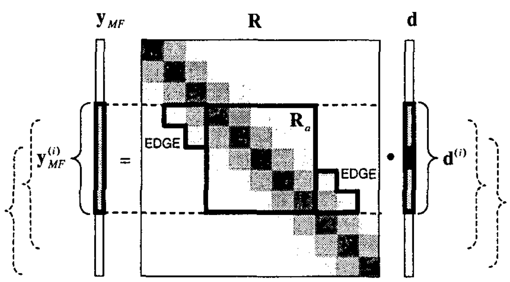

Fig. 2. Description of the vectors $\mathbf{y}_{MF}^{(i)}$, $\mathbf{d}^{(i)}$ e matrix $\mathbf{R}_a$ used in the SWD algorithm.

IV. Algorithm evaluation and comparison

Both SWD and ACD algorithms are compared in terms of computational complexity and performance. The key is the choice of the parameter a that should be large enough to neglect the edge effect that depends on the channel length (or equivalently on v).

Tables II, III and IV show the computational complexity of the approximate detectors in terms of real multiplications. The numerical values are expressed in terms of 10^5 multiplication units and refer to the case $K=8$, $v=5$, $N=61$, $Q=16$, $W=57$. The parameter s indicates the number of symbols that are estimated in each window of the SWD algorithm; details will be given later in this section. For sake of simplicity, we have used $M=1$ in the simulations as the factorization cost that involves the matrix $\mathbf{R}$ is almost independent of M (the number of antennas M affects mainly the size of matched filtering). Both ACD and SWD require the computation of $2NK$ samples of the S-T matched filter and the Cholesky factorization of the $aK \times aK$ matrix $\mathbf{R}_a$. The latter is carried out only once for the overall burst by the Schur algorithm, hence it requires $O(8K^3av)$ real multiplications.

TABLE II
Computational complexity comparision of the ACD and SWD methods.

Operation	Computational complexity
MF	$O(8QvMKN)$
Cholesky fact.	$O(8vK^3a)$
B/F SWD	$O(16vK^2a\frac{N}{s})$
B/F ACD	$O(16vK^2N)$

In the ACD, the large Cholesky factor is derived from $\mathbf{R}_a^{1/2}$ by copying the last column; then the overall system is solved by forward and backward substitution. The latter operations require $O(16vK^2N)$ real multiplications. Likewise, in the SWD for each window, the central symbols are detected for all K users by performing at first a forward substitution and then a partial backward substitution of dimension aK. The B/F processing of the $N-a+1$ windows requires $O(12vK^2aN)$ real multiplications. For $v << a << N$, the overall computational load of the SWD can be too expensive when compared with the ACD.

TABLE III
Computational complexity (in terms of real multiplication $\times 10^5$) of approximate detectors SWD and ACD. The parameters adopted are the same used in Table I.

a	s	Cholesky fact.		B/F	
		$M=1$	$M=8$	SWD	ACD
5	1	0.9	4.4	5.3	2.7
7	3	1.2	4.8	3.6	2.7
9	5	1.6	5.2	3.3	2.7
11	7	1.9	5.5	3.2	2.7
13	9	2.3	5.8	3.1	2.7
15	11	2.6	6.2	3.2	2.7

For this reason, we propose to modify the one-shot algorithm previously reviewed by detecting s symbols within each window ($1 \leq s \leq a$). At each step, the window slides s symbol intervals, therefore the number of windows decreases to $N_w = \lceil (N-(a-s))/s \rceil \propto N/s$ and the computational complexity reduces to $O(16vK^2a\frac{N}{s})$. In order to reduce the edge effects, the parameter s should be selected so that $a \geq s + 2(v-1)$ (note that ISI spans $v-1$ symbol intervals).

In Fig. 3 and 4, the performance of both ACD and SWD algorithms for MUD are compared in terms of BER for un-

TABLE IV

COMPUTATIONAL COMPLEXITY (IN TERMS OF REAL MULTIPLICATION $\times 10^5$) OF THE MUD ALGORITHMS SHOWN IN TABLE III. THE PARAMETERS ADOPTED ARE THE SAME USED IN TABLE II.

a	s	Total MUD			
		$M=1$		$M=8$	
		SWD	ACD	SWD	ACD
5	1	9.0	6.4	32.1	29.5
7	3	7.6	6.7	30.8	29.9
9	5	7.7	7.1	30.9	30.3
11	7	7.9	7.4	31.1	30.6
13	9	8.2	7.8	31.3	30.9
15	11	8.6	8.1	31.8	31.3

coded bits vs. E_b/N_0. The channel model adopted here is the COST-207 bad urban (BU) multipath propagation channel with 12 rays. From this example, it is sufficient to select $a = 9$ in the SWD algorithm to have negligible impacts on performance with respect to the ACD (characterized by $a = 5$) still requiring only about 20% more calculations (see Tables III and IV).

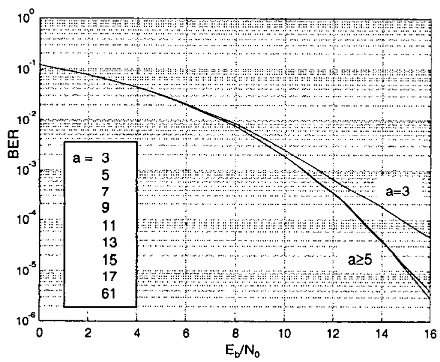

Fig. 3. Performance of MUD in terms of BER vs. SNR for ACD (BU environment, burst type 1, $N = 61$, $W = 57$, $Q = 16$, $K = 8$, $M = 1$, AWGN noise).

Fig. 5 and 6 show how the MUD algorithms approximate the exact detector in terms of the windowing parameter a. Both the bad urban (BU) and the typical urban (TU) environments are considered. As the TU channel has a lower delay spread ($5\mu s$, $v = 3$) than the BU ($10\mu s$, $v = 4$), it requires a smaller window size. Indeed Fig. 6 indicates that the approximation of the detector has no observable effects on the performance if $a = 3$ is selected for the ACD and $a = 7$ with $s = 3$ for the SWD.

It can be concluded from Fig. 3-6 that the SWD has a lower convergence speed with respect to the ACD. The SWD attempts to limit the memory length of the decorrelator to $a < N$ by approximating $\mathbf{R}^{-1}$ with a block band matrix with bandwidth a. While the SWD operates on the inverse filter, the ACD approximates the direct filter $\mathbf{R}^{1/2}$ and does not truncate the memory length of the decorrelator. It is clear that for the same size a, the edge effects introduced by the SWD are stronger than those due to the ACD, hence the SWD calls for a larger window size.

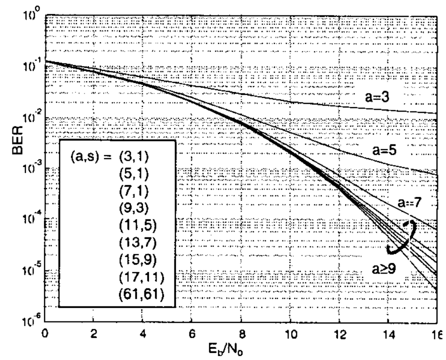

Fig. 4. Performance of MUD in terms of BER vs. SNR for SWD using the same parameter values of Fig.2 (the paramter s is specified in the frame).

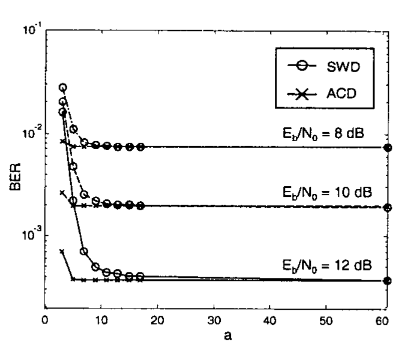

Fig. 5. Performance of MUD in terms of BER vs. a for ACD and SWD using the same parameter values of Fig.2 ($s = 1$)

In summary, to obtain the same performances, the SWD must adopt a larger windowing parameter a and requires a slightly greater computational complexity with respect to the ACD.

V. Systolic array

Exploiting the inherent parallelism of the SWD algorithm, a pipelined architecture can be implemented. The systolic architecture showed in Fig. 7 has been designed. We use an orthogonal array with $aK \times (aK+1)$ cells composed of three different processor types. The array is fed with the matrix $\mathbf{R}_a$ and the filtered vectors $\mathbf{y}_{MF}^{(i)}$ properly

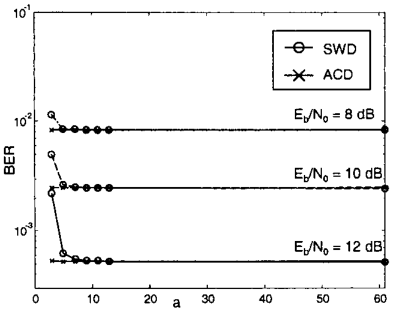

Fig. 6. Performance of MUD in terms of BER vs. a for ACD and SWD (TU environment, burst type 1, $N = 61$, $W = 57, Q = 16$, $K = 8$, $M = 1$, $s = 1$, AWGN noise).

interleaved and arranged to form the matrix $\mathbf{B}$ described in Fig. 8 for $s = 1$. Except for the first and last window, only Ks columns of the output matrix $\hat{\mathbf{D}}$ are necessary to estimate the vector $\hat{\mathbf{d}}$.

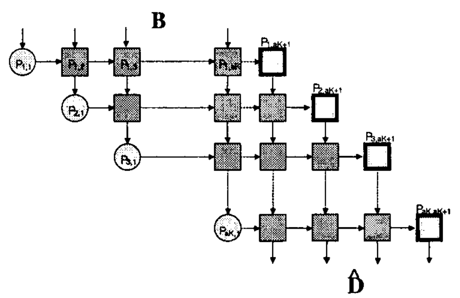

Fig. 7. Systolic array implementation of S-T MUD.

Both size and computation time of the circuit have been evaluated in terms of the algorithm parameters ($3 \leq a \leq 17, N = 61, W = 57, Q = 16, K = 8, 1 \leq s \leq 11, M = 8$). The number of cells $N_p = aK(aK+1)$ varies from 600 to 18632 while the number of clock cycles N_c required to process both semi-bursts increases linearly according to $N_c = 4aK + N_w - 2$. For instance, if $a = 7$, $s = 1$, it is $N_p = 3192$ and $N_c = 332$ clock cycles. Due to the low number of cycles and the high number of cells required, many architectural trade-offs can be exploited to keep the chip size at a reasonable level. The array design is being specified using VHDL while its FPGA implementation is currently under investigation.

In spite of the computational complexity disadvantage of the SWD algorithm with respect to the other methods previously described, its hardware implementation has several attractive properties that are summarized here: *i)* the possibility to easily implement an adaptive receiver; *ii)* the possibility to choose the order of processing for the

$$\mathbf{B} = \begin{bmatrix} \mathbf{Y}_{MF} \\ \mathbf{R}_a \end{bmatrix} = \begin{bmatrix} & & & y_{MF}^{(N_w)}(aK) \\ & & & \vdots \\ & & & y_{MF}^{(2)}(aK) \\ & y_{MF}^{(N_w)}(2) & \cdots & y_{MF}^{(1)}(aK) \\ y_{MF}^{(N_w)}(1) & \vdots & \cdots & R_a(aK,aK) \\ \vdots & y_{MF}^{(2)}(2) & \cdots & \vdots \\ y_{MF}^{(2)}(1) & y_{MF}^{(1)}(2) & \vdots & R_a(aK,2) \\ y_{MF}^{(1)}(1) & R_a(2,aK) & \vdots & R_a(aK,1) \\ R_a(1,aK) & \vdots & \vdots & - \\ \vdots & R_a(2,2) & \vdots & - \\ R_a(1,2) & R_a(2,1) & - & - \\ R_a(1,1) & - & - & - \end{bmatrix}$$

Fig. 8. Input matrix arrangement for the systolic implementation of Fig.7 when $s = 1$. Only one semi-burst is shown.

K users (e.g., assign a user priority); *iii)* the possibility to fully optimize the matrix inversion algorithm. In fact, due to the small dimensions of the $\mathbf{R}_a$ matrix, the array can be easily updated in real time using the channel estimation information. In addition, the algorithm used by the hardware system to compute $\mathbf{R}_a^{-1}$ may be optimized to match the required area-speed constraints. For instance, the proposed systolic array implements the Jordan diagonalization [7] in order to limit the use of multiplications and divisions.

VI. Conclusions

The comparison of two approximate methods for space-time MUD in TDD-UTRA system has shown that *i)* for S-T processing the cost of matched filter dominates but the it has a high degree of parallelism that must be exploited; *ii)* block-type Cholesky factorization is computationally more efficient than SWD in time-dispersive channels; *iii)* the SWD is the preferred choice in adaptive MUD when there is the need to track the channel variations within the bursts. *iv)* the SWD algorithm may be easily implemented with pipelined architectures that exploits its implicit parallelism.

References

[1] 1. J. Mayer, J. Schlee, T.Weber, *"Realtime feasibility of joint detection CDMA,"* Proc. EPMCC'97, pp. 245-252.

[2] M. Nicoli, U. Spagnolini, *"Adaptive-rank receiver for space-time DS CDMA systems,"* Proc. IEEE VTC 2001 Spring, Vol. 1, pp. 55.

[3] J. Rissanen, *"Algorithms for Triangular Decomposition of Block Hankel and Toeplitz Matrices with Application to Factoring Positive Matrix Polynomials,"* Math. Comp. Vol. 27, N. 121, pp. 147-154, Jan. 1973.

[4] 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.

[5] G.H. Golub, C.F. Van Loan, *"Matrix computations,"* The Johns Hopkins University Press, 1991.

[6] R. Lupas, S. Verdu, *"Near-far resistance of multiuser detectors in asynchronous channels,"* IEEE Trans. Commun., Vol. 38, pp. 496-508, Apr. 1990.

[7] P. Quinton, Y. Robert, *"Systolic Algorithms and Architectures,"* Prentice Hall, 1991.