# Efficient Joint Detection Techniques for TD-CDMA in the Frequency Domain

*Marius Vollmer*[1,2], *Jürgen Götze*[1], *Martin Haardt*[2]

1. Dept. of Electrical Engineering
Information Processing Lab
University of Dortmund
Otto-Hahn-Str. 4
D-44221 Dortmund, Germany
juergen.goetze@uni-dortmund.de
marius.vollmer@uni-dortmund.de

2. Siemens AG, ICN CA CTO 7
Mobile Communication
Chief Technical Office
Hofmannstr. 51
D-81359 Munich, Germany
Martin.Haardt@icm.siemens.de
Marius.Vollmer@icm.siemens.de

## Abstract

Third generation mobile radio systems will employ TD-CDMA in their TDD mode. To increase the capacity and performance of this system, the receiver will contain a *joint detector*. Joint detection is equivalent to solving a least squares problem, which represents a significant computational effort because of the amount of data that is involved. Therefore, algorithms and implementations must be developed that lower this complexity as far as possible without degrading the performance of the joint detector. This paper presents an algorithm that is based on the idea of extending the system matrix of the least squares problem to a block-circulant matrix. It is then possible to block-diagonalize the matrix with Fast Fourier Transforms. In addition, overlap-save techniques are presented that reduce the computational complexity further. The resulting algorithm is well suited for the implementation on parallel architectures.

## 1. Introduction

The TDD-Mode of 3rd generation mobile radio systems will be based on TD-CDMA [1]. To overcome the near/far problem of traditional CDMA systems, receiver structures have been proposed for TD-CDMA that perform joint (or multi-user) detection [2]. A joint detector combines the knowledge about all users that are active in one burst into one large system of equations. This knowledge consists of the channel impulse responses that have been estimated from training sequences, the spreading codes, and the received antenna samples. The resulting system of equations can be very large. In order to reduce the computational complexity, algorithms must be developed that exploit its special structural characteristics, namely its band and block-Toeplitz structure. In [3] an approach based on the Cholesky algorithm was presented. The band structure of the system matrix leads to an approximate block-Toeplitz structure in the desired Cholesky factor. This has been exploited by computing the Cholesky factor of a smaller subproblem and using it to build the complete Cholesky factor from copies of that smaller factor [4]. In [5] the Schur algorithm was used to exploit the Toeplitz structure directly. This approach leads to a row-oriented technique for approximating the Cholesky factor.

In this paper, we show a different approach for obtaining the solution of the joint detection problem. The original block-Toeplitz system matrix is extended into a block-circulant matrix. This block-circulant matrix can then be inverted with little computational effort by using block-FFTs and overlap-save techniques.

The paper is organized as follows: Section 2 explains the data model used to derive the system equation; section 3 shows how block-circulant matrices can be diagonalized by block-FFTs; section 4 applies this result to the TD-CDMA system; section 5 explains the overlap-save techniques; section 6 discovers the inherent parallelism of joint detection techniques in the frequency domain; finally, section 7 presents simulation results and computational complexity figures for several joint detection algorithms.

## 2. System Model

### 2.1. TD-CDMA

In the TD-CDMA system, $K$ CDMA codes are simultaneously active on the same frequency and in the same time slot. The different spreading codes allow the signal separation at the receiver. According to the required data rate, a given user might use several CDMA codes and/or time slots.

In Figure 1, the structure of one time slot is illustrated for the $k$th midamble and the $k$th spreading code. Here, $Q$ denotes the spreading factor of the data symbols and $N$ denotes the number of symbols in one data block. In this paper, we assume that all users use the same spreading factor. However, it is straightforward to extend the algorithms to variable spreading factors.

As explained in much more detail in [5], the transmission of one block of $N$ data symbols can be modelled by a system of linear equations,

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{V} & & & \\ & \boldsymbol{V} & & \\ & & \ddots & \\ & & & \boldsymbol{V} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \vdots \\ \boldsymbol{d}_N \end{bmatrix} + \boldsymbol{n} = \boldsymbol{T} \cdot \boldsymbol{d} + \boldsymbol{n}. \qquad (1)$$

In this equation, $\boldsymbol{x}$ is a vector containing the received samples from all $M$ antennas, $\boldsymbol{V}$ is a unstructured matrix that contains the knowledge about the estimated channel impulse responses and the spreading codes, $\boldsymbol{d}_i$ is a vector containing the $i$th transmitted symbol of all $K$ users, and the vector $\boldsymbol{n}$ represents temporal and spatial noise.

### 2.2. Joint Data Detection via Block Linear Equalization

Given the linear space-time data model in equation (1), we want to find a linear estimate of the $N$ data symbols transmitted by

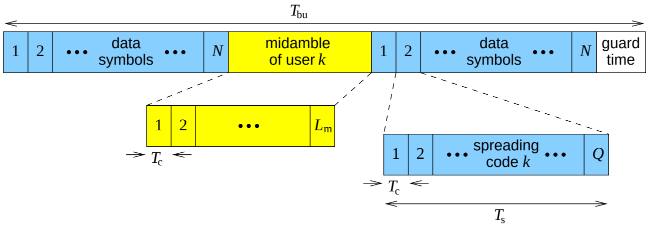


**Figure 1:** Time slot structure of the TD-CDMA system. Here, $T_{\rm bu}$, $T_{\rm s}$ $T_{\rm c}$, and $Q$ denote the burst duration, the symbol duration, the chip duration, and the spreading factor of the data symbols, respectively.

each of the $K$ users during the duration of one block (half burst), i.e., a *block linear equalizer*, such that

$$\hat{\boldsymbol{d}} = \boldsymbol{W}^H \boldsymbol{x} \in \mathbb{C}^{NK}. \tag{2}$$

One possible solution is to choose the space-time weighting matrix $\boldsymbol{W}^H \in \mathbb{C}^{(NK)\times M(NQ+W-1)}$ in (2) such that the Euclidean norm of the error

$$\|\boldsymbol{x} - \boldsymbol{T}\cdot\boldsymbol{d}\|_2^2$$

is minimized. It is given by the the standard least squares (LS) solution, where $\boldsymbol{W}^H$ is equal to the Moore-Penrose pseudo inverse (generalized inverse) of $\boldsymbol{T}$, i.e.,

$$\hat{\boldsymbol{d}} = \boldsymbol{W}^H\boldsymbol{x} = \boldsymbol{T}^+\boldsymbol{x} = \left(\boldsymbol{T}^H\boldsymbol{T}\right)^{-1}\boldsymbol{T}^H\boldsymbol{x}. \tag{3}$$

This LS solution does not take inter-cell interference into account, i.e., strong interferers from adjacent cells are not cancelled. To achieve this, one could employ the MVDR solution [6]. The MVDR solution can in turn be reduced to a LS solution by a preprocessing step.

## 3. Diagonalizing Block-Circulant Matrices

A circulant matrix is a square matrix where each column has the same elements as the column to the left of it, only rotated down by one position. For example,

$$\boldsymbol{C} = \begin{bmatrix} c_1 & c_4 & c_3 & c_2 \\ c_2 & c_1 & c_4 & c_3 \\ c_3 & c_2 & c_1 & c_4 \\ c_4 & c_3 & c_2 & c_1 \end{bmatrix}$$

is circulant. By extension, a matrix is said to be block-circulant if the entities denoted by $c_i$ above are not scalars but are themselves block-matrices. Since every $c_i$ appears once in each column and in each row, all $c_i$ must be block-matrices of the same size.

Due to the fact that the Fourier vectors are the eigenvectors of circulant matrices, systems of equations whose defining matrix is circulant can be solved efficiently in the frequency domain. The transformation to and from the frequency domain can be done efficiently with Fast Fourier Transforms [7]. This well known method can be extended to block-circulant systems.

When dealing with block structured matrices, it is convenient to use *colon notation*. For vectors, it is defined as follows

$$\begin{aligned} \boldsymbol{y} &= \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^T, \\ \boldsymbol{y}(i\!:\!j\!:\!k) &\equiv \begin{bmatrix} y_i & y_{i+j} & y_{i+2j} & \cdots & y_k \end{bmatrix}^T, \\ \boldsymbol{y}(i\!:\!k) &\equiv \boldsymbol{y}(i\!:\!1\!:\!k), \\ \boldsymbol{y}(i) &\equiv y_i, \\ \boldsymbol{y}(:) &\equiv \boldsymbol{y}. \end{aligned}$$

This notation is extended to matrices by applying the colon notation to row and column indices separately.

Just like circulant matrices can be diagonalized by (discrete) Fourier transforms, block-circulant matrices can be block-diagonalized by block Fourier transforms. Suppose the block-circulant matrix $\boldsymbol{C}$ has blocks of size $P\times K$, i.e., $c_i \in \mathbb{C}^{P\times K}$. Further, assume that $\boldsymbol{C}$ is of size $DP\times DK$, i.e., it has $D\times D$ blocks. Then we compute the matrix $\boldsymbol{\Lambda}$ as

$$\boldsymbol{\Lambda} = \boldsymbol{F}_{(P)}\boldsymbol{C}\boldsymbol{F}_{(K)}^{-1},$$

where $\boldsymbol{F}_{(P)}$ and $\boldsymbol{F}_{(K)}$ are block Fourier transform matrices with block sizes of $P\times P$ and $K\times K$ respectively. They are defined as

$$\boldsymbol{F}_{(n)} = \boldsymbol{F}\otimes\boldsymbol{I}_{(n)}, \tag{4}$$

where $\boldsymbol{I}_{(n)}$ is the identity matrix of size $n\times n$, $\boldsymbol{F}$ is the Fourier matrix of size $D\times D$ and $\otimes$ denotes the Kronecker product.

The resulting matrix $\boldsymbol{\Lambda}$ will be block-diagonal, that is, it will be of the form

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_1 & & & \\ & \boldsymbol{\Lambda}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{\Lambda}_D \end{bmatrix} \equiv \mathrm{diag}_{(P,K)}\left(\begin{bmatrix} \boldsymbol{\Lambda}_1 \\ \boldsymbol{\Lambda}_2 \\ \vdots \\ \boldsymbol{\Lambda}_D \end{bmatrix}\right)$$

where each $\boldsymbol{\Lambda}_i$ is a block-matrix of size $P\times K$. Therefore $\boldsymbol{\Lambda}$ is completely defined by the $\boldsymbol{\Lambda}_i$. Analogous to the scalar case, the $\boldsymbol{\Lambda}_i$ can be calculated as

$$\begin{bmatrix} \boldsymbol{\Lambda}_1 \\ \boldsymbol{\Lambda}_2 \\ \vdots \\ \boldsymbol{\Lambda}_D \end{bmatrix} = \boldsymbol{F}_{(P)}\boldsymbol{C}(:,1\!:\!K).$$

These properties of the block Fourier transform can be used to efficiently solve an equation like

$$\boldsymbol{C}\boldsymbol{d} = \boldsymbol{x} \tag{5}$$

where $\boldsymbol{C}$ is block-circulant. The LS solution $\hat{\boldsymbol{d}} = \boldsymbol{C}^+\boldsymbol{x}$ of (5) can be computed as

$$\hat{\boldsymbol{d}} = \boldsymbol{F}_{(K)}^{-1}\boldsymbol{\Lambda}^+\boldsymbol{F}_{(P)}\boldsymbol{x}.$$

Thus, we need to apply a block Fourier transformation to $\boldsymbol{x}$, then multiply by the pseudo-inverse of the block-diagonal matrix $\boldsymbol{\Lambda}$ and finally calculate an inverse block Fourier transform to get the result $\hat{\boldsymbol{d}}$. Since $\boldsymbol{\Lambda}$ is block-diagonal, multiplying by its pseudo-inverse is in general significantly cheaper than using $\boldsymbol{C}$ directly.

## 4. Application to TD-CDMA

Although $\boldsymbol{T}$ in (3) is not block-square, the block-columns of $\boldsymbol{T}$ are already rotated versions of the first block-column. Therefore, we can simply add block columns to it until it is block-square. The number of block-columns that we must add depends on the degree of the block-band structure of $\boldsymbol{T}$, which is the same as the inter-symbol interference in the original transmission system. Figure 2 shows how $\boldsymbol{T}$ can be extended to be block-circulant.

After $\boldsymbol{T}$ has been extended to be block-circulant, it has $D\times D$ blocks of size $P\times K$, where $D = N + \lceil(Q+W-1)/Q\rceil - 1$

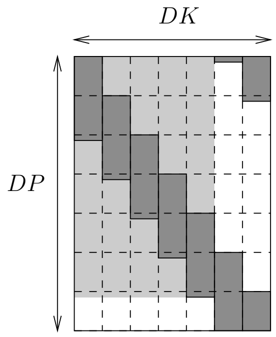


**Figure 2:** Extending the block Sylvester matrix $\boldsymbol{T}$ (light shaded part) to be block-circulant.

and $P = MQ$.[1] The vector $\boldsymbol{x}$ needs to be zero-padded at its end so that it has length $DP$. Likewise, the new solution vector contains the desired results in its first $NK$ elements.

Note that only the first block-column of $\boldsymbol{T}$ is needed to compute $\boldsymbol{\Lambda}$. Therefore, we just need to extend $\boldsymbol{V}$ with zero-filled rows such that it is of size $DP \times K$. Let $\mathbf{0}_{(n)}$ denote the zero vector of length $n$ and $\mathbf{0}_{(n,m)}$ the zero matrix of size $n \times m$. We can then summarize the steps required to get a least squares estimate $\hat{\boldsymbol{d}}$ of $\boldsymbol{d}$ as follows:

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}^T & \mathbf{0}_{(DP-M(NQ+W-1))} \end{bmatrix}^T$$
$$\boldsymbol{\Lambda} = \mathrm{diag}_{(P,K)} \left( \boldsymbol{F}_{(P)} \begin{bmatrix} \boldsymbol{V}^T & \mathbf{0}_{(K,DP-M(Q+W-1))} \end{bmatrix}^T \right)$$
$$\tilde{\boldsymbol{d}} = \boldsymbol{F}_{(K)}^{-1} \boldsymbol{\Lambda}^+ \boldsymbol{F}_{(P)} \tilde{\boldsymbol{x}}$$
$$\hat{\boldsymbol{d}} = \tilde{\boldsymbol{d}}(1{:}NK)$$

Simulations (Figure 6) have shown that the error that occurs by solving for $\hat{\boldsymbol{d}}$ with this extended version of $\boldsymbol{T}$ is insignificant. This is due to the fact the distortions introduced by making $\boldsymbol{T}$ block-square affect only the guard periods between bursts. Neglecting the noise, the guard period between bursts makes it possible to model the channel correctly with a block-circulant matrix.

## 5. Overlap-Save

The convolution matrix $\boldsymbol{T}$ with its strong band structure offers possibilities to reduce the computational demands of the joint detector even further.

The idea is to reduce the size of the involved matrices and solve the whole problem by solving multiple smaller ones instead. The reduction in size is expressed by forcing $D$ to smaller values when deriving a block-circular matrix from $\boldsymbol{T}$ according to Figure 2. With such a smaller matrix, only a smaller part of the data vector can be estimated of course; thus, we need to partition the data vector into slices of length $DK$. But if $D$ is smaller than its ideal value $N + \lceil (Q + W - 1)/Q \rceil - 1$, the distortions mentioned in the previous section do no longer fall into the guard period, leading to unacceptable errors in the estimated data vector slices.

Figure 3 depicts this effect. The dashed line shows the relative error of the data symbols of one user where $D$ has been set to 32 and the full vector of $N = 69$ symbols has been calculated by carrying out the frequency domain detection three times on successive blocks of 32 symbols each. Obviously, each run of $D = 32$ symbols has large errors at the beginning and the end, but not in its middle part.

[1] The notation $\lceil x \rceil$ denotes the *ceiling* of $x$, i. e., the smallest integer that is greater than or equal to $x$.

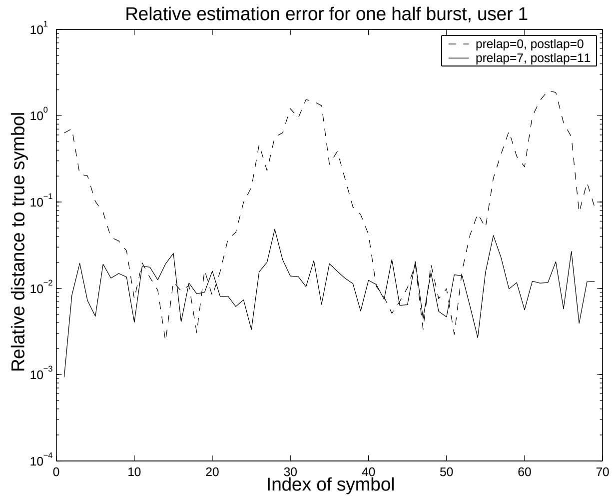


**Figure 3:** Estimation errors when using overlap-save techniques. $Q = 16$, $W = 57$, $N = 69$, $K = 4$, $M = 1$, $D = 32$, no noise.

As a remedy, one discards a certain number of symbols at the start of the data vector slice (the *prelap*) and at the end of it (the *postlap*). The computation of the complete data vector needs to be arranged in such a way that the discarded symbols from the previous slice can be found in the middle of the next one as depicted in Figure 4. Figure 3 shows that the relative error for such an overlapping computation has been reduced to a lower level for all symbols (solid line).

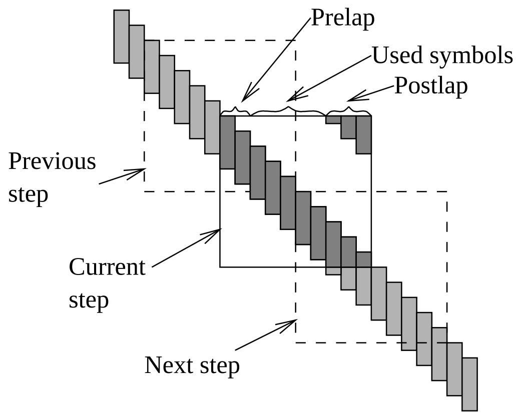


**Figure 4:** Overlapping Convolution Matrices with $D = 10, p^- = 2, p^+ = 3$

Using the colon notation defined above, we can describe the overlapping process more formally. Let us denote the amount of prelap by $p^-$ and the amount of postlap by $p^+$. Figure 5 shows the pseudo-code for performing the described overlap save technique.

## 6. Parallelism

Besides finding algorithms that require as few operations as possible for solving a certain problem, it is also important to engineer the algorithms in such a way that many of these operations

$$
\begin{aligned}
&p \leftarrow D - p^- - p^+ \\
&S \leftarrow \lceil N/p \rceil \\
&\tilde{\boldsymbol{x}} \leftarrow \begin{bmatrix} \mathbf{0}_{(p^-)} & \boldsymbol{x}^T & \mathbf{0}_{(Sp+p^+-M(NQ+W-1))} \end{bmatrix}^T \\
&\boldsymbol{\Lambda} \leftarrow \mathrm{diag}_{(P,K)} \left( \boldsymbol{F}_{(P)} \begin{bmatrix} \boldsymbol{V}^T & \mathbf{0}_{(K,DP-M(Q+W-1))} \end{bmatrix}^T \right) \\
&\textbf{for } s = 1 : S \\
&\quad \boldsymbol{x}^{(s)} \leftarrow \boldsymbol{x}((s-1)pP+1 : ((s-1)p+D)P) \\
&\quad \boldsymbol{d}^{(s)} \leftarrow \boldsymbol{F}_{(K)}^{-1} \boldsymbol{\Lambda}^+ \boldsymbol{F}_{(P)} \boldsymbol{x}^{(s)} \\
&\quad \hat{\boldsymbol{d}}((s-1)pK+1 : spK) \leftarrow \boldsymbol{d}^{(s)}(p^- : p^- + p) \\
&\textbf{end}
\end{aligned}
$$

**Figure 5:** Using overlapping blocks to compute $\hat{\boldsymbol{d}}$

can be easily performed in parallel on the target architecture. The joint detection algorithm described in the previous sections offers many such opportunities.

The execution of the algorithm can be divided into three stages, with a lot of parallelism in each stage. Furthermore, if dedicated hardware is available for each stage, they can be pipelined so that all stages can be performed in parallel, too. The following takes into account that one burst consists of two half bursts.

The first stage consists of all block-FFT operations, namely the computation of $\boldsymbol{\Lambda}$ and $\boldsymbol{F}_{(P)}\boldsymbol{x}^{(s)}$, $1 \le s \le S$. Clearly, all transformations are independent and can thus be computed in parrallel. Moreover, it can be seen from the definition of $\boldsymbol{F}_{(P)}$ in (4), that there also exists parallelism within each transformation.

With the colon notation, the block-Fourier transformation can be taken apart as follows. Given

$$
\boldsymbol{y} \in \mathbb{C}^{DP}, \quad \boldsymbol{F} \in \mathbb{C}^{D \times D}, \quad \boldsymbol{I}_{(P)} \in \mathbb{C}^{P \times P}
$$
$$
\boldsymbol{F}_{(P)} = \boldsymbol{F} \otimes \boldsymbol{I}_{(P)}, \quad \boldsymbol{z} = \boldsymbol{F}_{(P)} \boldsymbol{y}
$$

we have

$$
\boldsymbol{z}(i\!:\!P\!:\!n-P+i) = \boldsymbol{F}\boldsymbol{y}(i\!:\!P\!:\!n-P+i) \quad 1 \le i \le P.
$$

As can be seen, a block-FFT of block-size $P$ decomposes into $P$ parallel non-block-FFTs. Therefore, the first stage consists of $P(K+2S)$ parrallel FFTs of length $D$.

The second stage has to compute $\boldsymbol{\Lambda}^+ \boldsymbol{F}_{(P)} \boldsymbol{x}^{(s)}$. Since $\boldsymbol{\Lambda} \in \mathbb{C}^{DP \times DK}$ is block-diagonal, this problem decomposes into $D$ independent systems of linear equations, with $2S$ right hand sides each.

The third stage finally has to apply block-IFFTs to $\boldsymbol{F}_{(K)}\boldsymbol{d}^{(s)}$ to yield $\boldsymbol{d}^{(s)}$. This can be done with $2SK$ independent inverse FFTs of length $D$, analogous to the first stage.

## 7. Simulation Results and Computational Complexity

Figure 6 compares the frequency domain approach for different overlapping degrees with the true least squares solution, obtained via a Cholesky factorization of $\boldsymbol{T}^H\boldsymbol{T}$. For white noise, the least squares solution corresponds to the zero forcing block linear equalizer.

The simulation scenario includes $K = 8$ users. To investigate the near/far resistance of the joint detector, four of these users have a power that is 20 dB above the remaining two users. This corresponds to a severe near/far scenario. The bit error ratio shown in Figure 6 is the mean of the four weaker users. It shows that the presented joint detector is able to handle this critical situation better than the approximated Cholesky joint detector. The receiver has $M = 1$ antenna. For reference, figure 7 shows the mean of the bit error ratios of all 8 users when they all have the same power.

The label "C" in the legend denotes the joint detector based on the Cholesky decomposition [3], "F" denotes the algorithm presented in this paper. "C exact" is the exact Cholesky decomposition and represents the true least squares solution. "C row" denotes a row-wise approximation of the Cholesky factors were 2 block-rows have been computed [5]. It yields a slightly worse bit error ratio performance as "C exact". "C tri" uses the triangle approximation described in [4] with a sub-matrix of $2 \times 2$ blocks. It has a much worse performance than "C exact". The parameters for the Fourier algorithms are presented in the legend as $D/p^-/p^+$. The performance of the Fourier detector with no overlap ("F -/-/-") is identical to the true least squares solution "C exact". When using significant overlap ("F 32/3/5"), the performance is still indistinguishable from "C exact" and using little overlap ("F 16/1/2") the performance is still comparable to "C row".

In figure 7, "C row" and "C tri" show the same performance. It is slightly worse than the performance of the remaining algorithms, which is about the same.

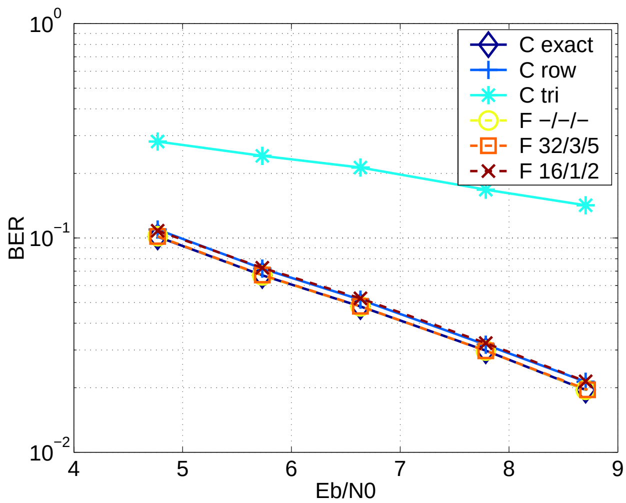


**Figure 6:** Bit error ratios after error correction of joint detection using different algorithms. The system parameters were chosen as $Q = 16$, $W = 57$, $N = 69$, $K = 8$, $M = 1$. The channel has been modeled as type Vehicular/A with a mobile velocity of 120 km/h. See the text for a discussion of the results.

Figures 8, 9 and 10 compare the computational complexity of the joint detection algorithms that have been used in the simulations of Figure 6. It can be seen that the frequency domain approach with overlap has a lower computational complexity than the approximated Cholesky approach.

## 8. Conclusions

In this paper, we have shown that performing joint detection in the frequency domain has a lower computational complexity than the approximated Cholesky joint detector while attaining a better bit-error ratio performance. The low computational complexity has been achieved by using overlap-save techniques for the deconvolution. Overlapping was performed both at the beginning and at the end of the considered data vector.

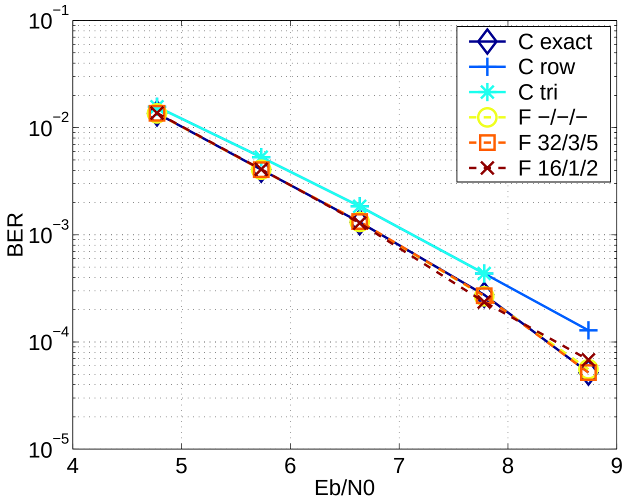


**Figure 7:** Same scenario as in figure 6, but all users are received with the same power.

Both the block-FFTs and the inversion of the diagonalized system matrix consist of independent subproblems that can be solved in parallel.

It remains to be investigated how these techniques apply to W-CDMA systems.

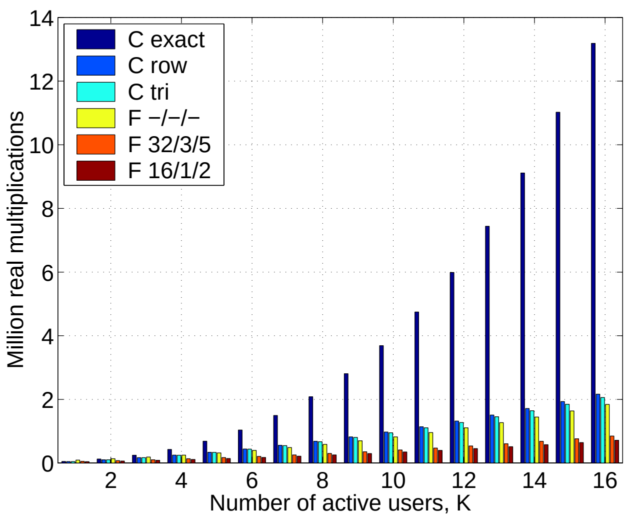


**Figure 8:** Computational complexity of the joint detection algorithms, as a function of the number of active codes, $K$. $Q = 16$, $W = 57$, $N = 69$, $M = 1$.

## References


[1] M. Haardt and W. Mohr, "The complete solution for third generation mobile communications: Two modes on air - one winning strategy," in *Proc. 3G Wireless*, 2000.

[2] P. Jung and J. J. Blanz, "Joint detection with coherent receiver antenna diversity in CDMA mobile radio systems," *IEEE Trans. on Vehicular Technology*, vol. 44, pp. 76–88, 1995.

[3] J. Blanz, A. Klein, M. Naßhan, and A. Steil, "Performance of a Cellular Hybrid C/TDMA Mobile Radio System Applying Joint Detection and Coherent Receiver Antenna Diversity," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 4, May 1994.

[4] J. Mayer, J. Schlee, and T. Weber, "Realtime feasibility of joint detection CDMA," in *Proc. 2nd European Personal Mobile Communications Conference*, Bonn, Germany, Sept. 1997, pp. 245–252.


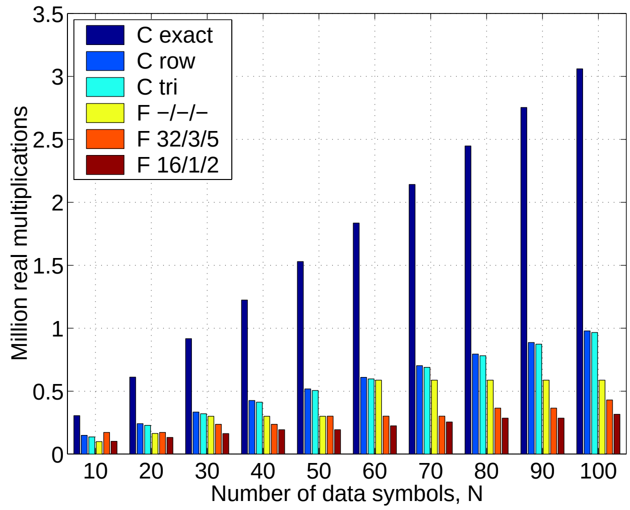


**Figure 9:** Computational complexity of the joint detection algorithms, as a function of the number of data symbols, $N$. $Q = 16$, $W = 57$, $K = 8$, $M = 1$.

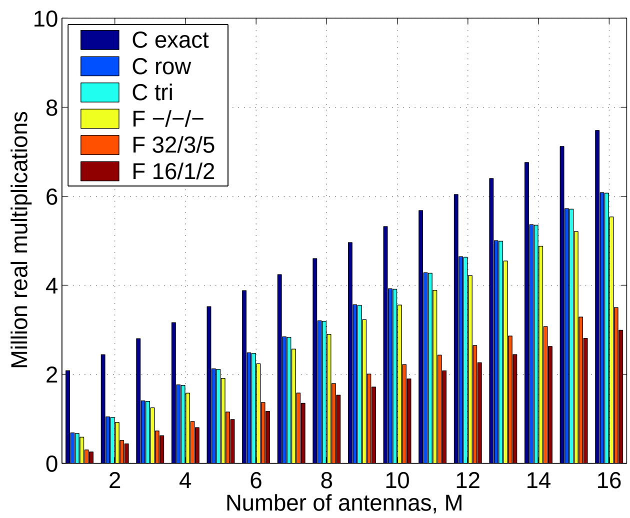


**Figure 10:** Computational complexity of the joint detection algorithms, as a function of the number of antennas, $M$. $Q = 16$, $W = 57$, $N = 69$, $K = 8$.


[5] M. Vollmer, M. Haardt, and J. Götze, "Schur algorithms for joint detection in TD-CDMA based mobile radio systems," *Annals of Telecommunications (special issue on multi user detection)*, vol. 54, no. 7-8, pp. 365–378, July-August 1999.

[6] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley and Sons, New York, NY, 1969.

[7] G.H. Golub and C.F. van Loan, *Matrix Computations*, The John Hopkins University Press, third edition, 1996.

[8] G. Ammar and W.B. Gragg, "Superfast Solution of Real Positive Definite Toeplitz Systems," *SIAM J. Matrix Anal. Appl.*, vol. 9, pp. 61–76, 1988.

[9] B. Steiner and P. Jung, "Optimum and suboptimum channel estimation for the uplink of CDMA mobile radio systems with joint detection," *European Trans. on Telecommunications and Related Techniques*, vol. 5, pp. 39–50, 1994.