Efficient Joint Detection Techniques in the Frequency Domain for Third Generation Mobile Radio Systems

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
goetze@dt.e-technik.uni-dortmund.de
mvo@dt.e-technik.uni-dortmund.de

2. Siemens AG, ICN CA CTO 7
Communication on Air
Chief Technical Office
Hofmannstr. 51
D-81359 Munich, Germany
Martin.Haardt@icn.siemens.de
Marius.Vollmer@icn.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 much 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 by 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. It has a lower computational complexity than existing methods while yielding a better bit error ratio performance.

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 present 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; section 7 takes a closer look at the FFT algorithm itself by optimizing it for our applications; finally, section 8 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 kth midamble and the kth 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.

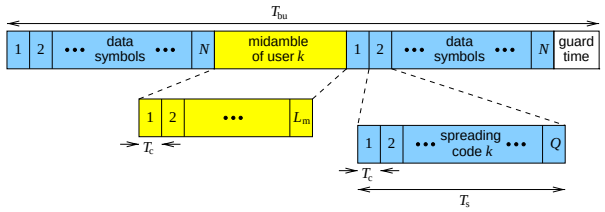


Figure 1: Time slot structure of the TD-CDMA system. Here, T_{bu}, T_{s}, T_{c}, and Q denote the burst duration, the symbol duration, the chip duration, and the spreading factor of the data symbols, respectively.

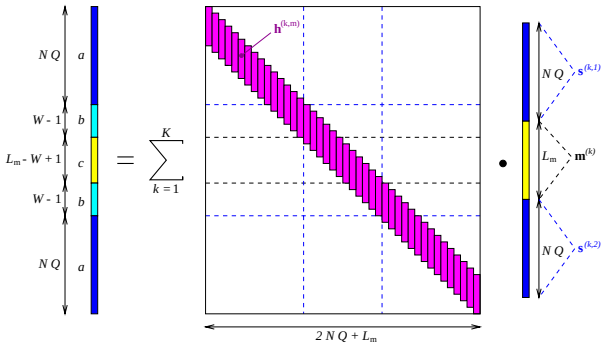


Figure 2: Received measurements at the mth antenna during the duration of one time slot. In this illustration, the additive noise and inter-cell-interference are not considered. The parts denoted by $\boldsymbol{s}^{(k,1)}$ and $\boldsymbol{s}^{(k,2)}$ contain the spread data symbols of the two half bursts of user k and $\boldsymbol{m}^{(k)}$ contains the midamble of user k.

2.2. Data Model

The received measurements at the mth antenna during the duration of one time slot are illustrated in Figure 2. The parts of the received measurement vector that are marked by the letter *a* are only influenced by the corresponding data blocks, the part that is marked by *c* is exclusively determined by the transmitted midambles, and the parts of the measurement vector marked by *b* are influenced by the transmitted midambles and the corresponding data blocks.

The channel impulse response (CIR) vectors between the kth mobile and the mth antenna $\boldsymbol{h}^{(k,m)} \in \mathbb{C}^W$ are estimated from the part of this received measurement vector that is marked by *c* as, for instance, described in [6].

Let us combine the N data symbols $d_n^{(k)}$, $1 \le n \le N$, that are transmitted on the kth spreading code during one data block (half burst) to the vector

$$\boldsymbol{d}^{(k)} = \begin{bmatrix} d_1^{(k)} & d_2^{(k)} & \cdots & d_N^{(k)} \end{bmatrix}^T \in \mathbb{C}^N. \qquad (1)$$

The kth spreading code consists of Q complex chips $c_q^{(k)}$, $1 \le q \le Q$, and is denoted as

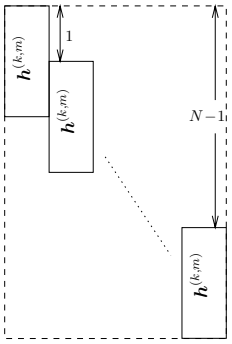
$$\boldsymbol{c}^{(k)} = \begin{bmatrix} c_1^{(k)} & c_2^{(k)} & \cdots & c_Q^{(k)} \end{bmatrix}^T \in \mathbb{C}^Q, \quad 1 \le k \le K.$$

With this definition, the block-diagonal spreading matrix $\boldsymbol{C}^{(k)}$ that corresponds to the kth code can be written as

$$\boldsymbol{C}^{(k)} = \boldsymbol{I}_{(N)} \otimes \boldsymbol{c}^{(k)} \in \mathbb{C}^{NQ \times N},$$

where $\boldsymbol{I}_{(N)}$ is the identity matrix of size $N \times N$ and $\otimes$ denotes the Kronecker product. Assume that K spreading codes are transmitted at the same time. After eliminating the influence of the midamble, cf. Figure 2, the received measurements at the mth antenna obey the following linear model:

$$\boldsymbol{x}^{(m)} = \begin{bmatrix} x_1^{(m)} & x_2^{(m)} & \cdots & x_{NQ+W-1}^{(m)} \end{bmatrix}^T \quad (1 \le m \le M)$$



$$= \sum_{k=1}^{K} \cdot \boldsymbol{C}^{(k)} \cdot \boldsymbol{d}^{(k)} + \boldsymbol{n}^{(m)}$$

$$= \sum_{k=1}^{K} \boldsymbol{H}^{(k,m)} \boldsymbol{C}^{(k)} \boldsymbol{d}^{(k)} + \boldsymbol{n}^{(m)}$$

$$= \sum_{k=1}^{K} \boldsymbol{B}^{(k,m)} \boldsymbol{d}^{(k)} + \boldsymbol{n}^{(m)} \qquad (2)$$

with $\boldsymbol{B}^{(k,m)} = \boldsymbol{H}^{(k,m)} \boldsymbol{C}^{(k)}$.

The matrix $\boldsymbol{H}^{(k,m)} \in \mathbb{C}^{(NQ+W-1)\times(NQ)}$ is a Toeplitz matrix that contains estimates of the channel impulse response (CIR) vector $\boldsymbol{h}^{(k,m)} \in \mathbb{C}^W$ of the kth user (corresponding to the kth spreading code) at the mth antenna. The vector $\boldsymbol{n}^{(m)}$ represents the noise and interference received at antenna m and $\boldsymbol{B}^{(k,m)}$ is a block-Toeplitz matrix consisting of combined CIR vectors $\boldsymbol{b}^{(k,m)}$ that can be expressed as the convolution of the channel impulse response (CIR) vector $\boldsymbol{h}^{(k,m)}$ with the corresponding spreading code $\boldsymbol{c}^{(k)}$, i.e.,

$$\boldsymbol{b}^{(k,m)} = \boldsymbol{h}^{(k,m)} * \boldsymbol{c}^{(k)} \in \mathbb{C}^{Q+W-1}.$$

Using the definition of $\boldsymbol{d}^{(k)} \in \mathbb{C}^N$ in (1), the transmitted data symbols of all K users are combined in the following

fashion,

$$\boldsymbol{d} = \text{vec}\left\{\begin{bmatrix}\boldsymbol{d}^{(1)} & \boldsymbol{d}^{(2)} & \cdots & \boldsymbol{d}^{(K)}\end{bmatrix}^T\right\} \in \mathbb{C}^{NK}.$$

Here and in the sequel, the vec$\{\cdot\}$ operator is defined according to

$$\text{vec}\left\{\begin{bmatrix}a_1 & a_2 & \cdots \\ b_1 & b_2 & \cdots\end{bmatrix}\right\} = \begin{bmatrix}a_1 & b_1 & a_2 & b_2 & \cdots\end{bmatrix}^T,$$

that is, the vec$\{\cdot\}$ operator forms a column vector from the elements of its argument matrix by concatenating the columns of that matrix, starting by the left. The combined CIR vectors $\boldsymbol{b}^{(k,m)}$ of the kth user at all M antennas can also be simplified to a single combined CIR vector in the space-time domain,

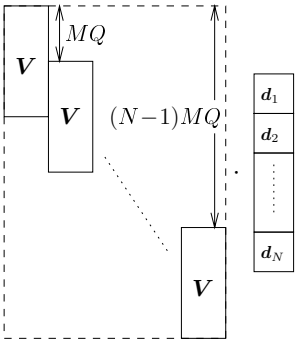
$$\boldsymbol{b}^{(k)} = \text{vec}\left\{\begin{bmatrix}\boldsymbol{b}^{(k,1)} & \cdots & \boldsymbol{b}^{(k,M)}\end{bmatrix}^T\right\} \in \mathbb{C}^{M(Q+W-1)}.$$

Moreover, let us define the space-time array measurement vector (during one half burst) as

$$\boldsymbol{x} = \text{vec}\left\{\begin{bmatrix}\boldsymbol{x}^{(1)} & \cdots & \boldsymbol{x}^{(M)}\end{bmatrix}^T\right\} \in \mathbb{C}^{M(NQ+W-1)}.$$

Using equation (2), the vector $\boldsymbol{x}$ may then be expressed as

$$\boldsymbol{x} = \begin{bmatrix}\boldsymbol{V} & & & \\ & \boldsymbol{V} & & \\ & & \ddots & \\ & & & \boldsymbol{V}\end{bmatrix} \cdot \begin{bmatrix}\boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \vdots \\ \boldsymbol{d}_N\end{bmatrix} + \boldsymbol{n} = \boldsymbol{T}\cdot\boldsymbol{d} + \boldsymbol{n}, \quad (3)$$



where the matrix

$$\boldsymbol{V} = \begin{bmatrix}\boldsymbol{b}^{(1)} & \boldsymbol{b}^{(2)} & \cdots & \boldsymbol{b}^{(K)}\end{bmatrix} \in \mathbb{C}^{M(Q+W-1)\times K}$$

contains the combined CIR vectors of all K users in the space-time domain. In equation (3), the space-time vector

$$\boldsymbol{n} = \text{vec}\left\{\begin{bmatrix}\boldsymbol{n}^{(1)} & \cdots & \boldsymbol{n}^{(M)}\end{bmatrix}^T\right\} \in \mathbb{C}^{M(NQ+W-1)}.$$

models inter-cell interference, i.e., dominant interferers from adjacent cells, and additive (thermal) noise. Notice that the matrix $\boldsymbol{T} \in \mathbb{C}^{M(NQ+W-1)\times(NK)}$ in equation (3) has a block-Toeplitz structure, which we are going to exploit to derive an efficient data detection algorithm.

2.3. Joint Data Detection via Block Linear Equalization

Given the linear space-time data model in equation (3), we want to find a linear estimate of the N data symbols transmitted by each of the K users during the duration of one block (half burst), i.e., a *block linear equalizer*, such that

$$\hat{\boldsymbol{d}} = \boldsymbol{W}^H\boldsymbol{x} \in \mathbb{C}^{NK}. \quad (4)$$

One possible solution is to choose the space-time weighting matrix $\boldsymbol{W}^H \in \mathbb{C}^{(NK)\times M(NQ+W-1)}$ in (4) 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.,

$$\hat{\boldsymbol{d}} = \boldsymbol{W}^H\boldsymbol{x} = \boldsymbol{T}^+\boldsymbol{x} = \left(\boldsymbol{T}^H\boldsymbol{T}\right)^{-1}\boldsymbol{T}^H\boldsymbol{x}. \quad (5)$$

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 [7]. 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 inverse Fourier vectors are eigenvectors of circulant matrices, a system 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 [8]. In this paper, this well known method is 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

$$\boldsymbol{F}_{(n)} = \boldsymbol{F} \otimes \boldsymbol{I}_{(n)}, \tag{6}$$

where $\boldsymbol{F}$ is the Fourier matrix of size $D \times D$.

The resulting matrix $\boldsymbol{\Lambda}$ will be block-diagonal, i.e., it can be expressed as

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

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 by aplying the block Fourier transform to the first block-column of $\boldsymbol{C}$, i.e,

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

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

$$\boldsymbol{C}\boldsymbol{d} = \boldsymbol{x} \tag{7}$$

where $\boldsymbol{C}$ is block-circulant. The LS solution $\hat{\boldsymbol{d}} = \boldsymbol{C}^+ \boldsymbol{x}$ of (7) is given by

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

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 (5) is not block-square, the block-columns of $\boldsymbol{T}$ are already rotated versions of the first block-column. Therefore, we can simply complete its last block-row and 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 3 shows how $\boldsymbol{T}$ can be extended to be block-circulant.

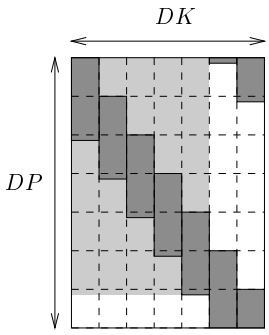


Figure 3: Extending the block Sylvester matrix $\boldsymbol{T}$ (light shaded part) to be block-circulant.

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

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

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 9) 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 extending $\boldsymbol{T}$ into a block-square matrix affect only the guard periods between bursts. Neglecting the noise, the guard period between bursts facilitates the modelling of the channel by 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 3. 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.

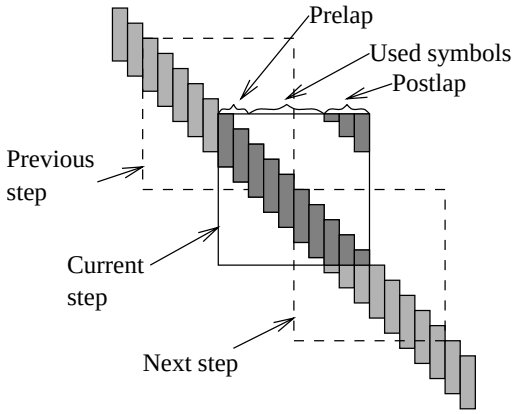


Figure 4: Overlapping Convolution Matrices with $D = 10, p^- = 2, p^+ = 3$

Figure 5 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.

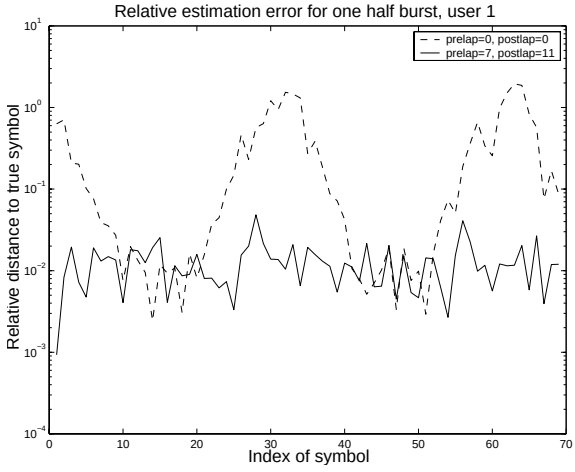


Figure 5: Estimation errors when using overlap-save techniques. $Q = 16$, $W = 57$, $N = 69$, $K = 4$, $M = 1$, $D = 32$, no noise.

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 5 shows that the relative error for such an overlapping computation has been reduced to a lower level for all symbols (solid line).

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 6 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 can easily be performed in parallel. The joint detection algorithm described in the previous sections offers many such opportunities. This enables the real-time solution of systems of equations with larger dimensions.

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

Figure 6: Using overlapping blocks to compute $\hat{\boldsymbol{d}}$. The prelap and postlap is denoted by p^- and p^+, respectively. S is the number of steps required.

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 described procedure 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 $\boldsymbol{F}_{(P)}\boldsymbol{x}^{(s)}$, $1 \le s \le S$ (see Figure 6. 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 (6), that there also exists parallelism within each transformation.

With the colon notation, the block-Fourier transformation of an arbitrary column vector $\boldsymbol{y}$ 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 conventional FFTs. Therefore, the first stage consists of $P(K + 2S)$ parallel 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. Sub-FFT Optimizations

The computational complexity of the well-known FFT algorithm [9] can be further reduced if the transformed vector

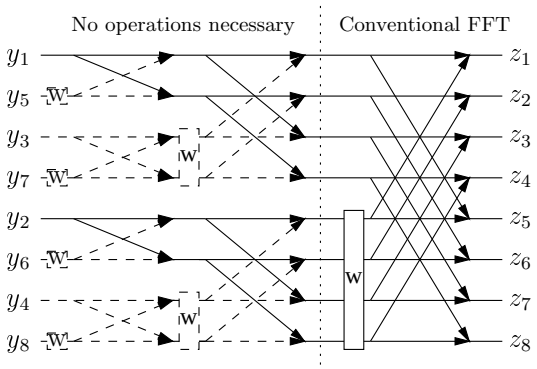


Figure 7: Flow graph of 8-point decimation-in-time FFT with band structured data. Only y_1 and y_2 are non-zero (solid lines), thus $m = 2$.

consists mostly of zeros with only a few non-zero elements at its beginning. Let $\boldsymbol{y} \in \mathbb{C}^n$ with $\boldsymbol{y}(i) = 0$ for $i > m$ where both n and m are powers of two. In that case, the number of multiplications for a radix-2, decimation-in-time FFT can be reduced from $\frac{n}{2}\log_2 n$ to $\frac{n}{2}\log_2 m$. In our application we use this optimization for computing $\boldsymbol{\Lambda}$ where we have $n = \lceil D \rceil_2$ and $m = \lceil (Q + W - 1)/Q \rceil_2$ which results in significant savings for typical cases.[2]

The way to achieve this reduction in computational complexity is to stop the recursive decent of the Cooley-Tukey FFT algorithm earlier. This can be done if the vector that is to be transformed is known to contain only one non-zero element, because the transformation of such a vector is trivial. This situation is reached after $\log_2 m$ subdividing steps. Figure 7 depicts this reasoning for $n = 8$ and $m = 2$. Figure 8 shows the pseudo-code for a possible implementation. The function `bitreverse` performs the usual reshuffling of the input vector that is needed for decimation-in-time FFT algorithms.

8. Simulation Results and Computational Complexity

Figure 9 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 with one code each. To investigate the near/far resistance of the joint detector, four of these users have a power that is 20 dB above the remaining four users. This corresponds to a severe near/far scenario. The bit error ratio shown in Figure 9

[2]The notation $\lceil x \rceil_k$ extends the definition in footnote 1. It denotes the smallest integer greater or equal to x that is a power of k.

Purpose: Compute $\boldsymbol{z} = \boldsymbol{F}\boldsymbol{y}$, where $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{C}^n$ and $\boldsymbol{y}(i) = 0$ for $i > m$. Both n and m are powers of two.

$\boldsymbol{z} \leftarrow$ `bitreverse`$(\boldsymbol{y})$ *(see text)*
$k \leftarrow n/m$
for $l = 1 : k : n - k + 1$
  for $i = 1 : k - 1$
    $\boldsymbol{z}(l + i) \leftarrow \boldsymbol{z}(j)$
  end
end
while $k < n$
  for $l = 0 : k - 1$
    for $i = l + 1 : 2k : n - 2k + l + 1$
      $t \leftarrow \boldsymbol{z}(i + k)e^{-j\pi l/k}$
      $\boldsymbol{z}(i + k) \leftarrow \boldsymbol{z}(i) - t$
      $\boldsymbol{z}(i) \leftarrow \boldsymbol{z}(i) + t$
    end
  end
  $k \leftarrow 2k$
end

Figure 8: Decimation-in-time, radix-2 FFT algorithm for band-structured data

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.

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 2×2 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".

Figures 10, 11 and 12 compare the computational complexity of the joint detection algorithms that have been used in the simulations of Figure 9. It can be seen that the frequency domain approach with overlap has a lower computational complexity than the approximated Cholesky approach.

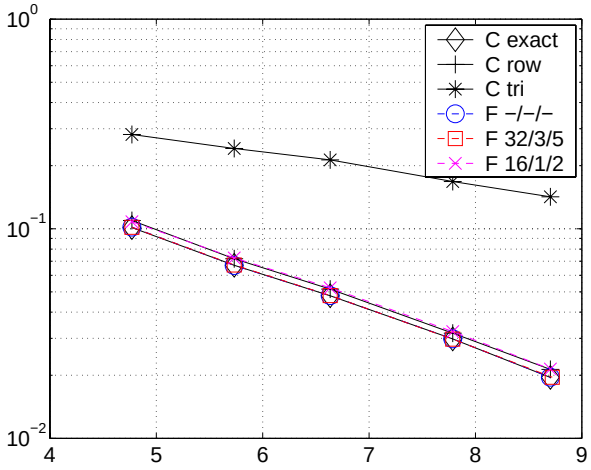


Figure 9: Bit error ratios after error correction of joint detection using different algorithms. The system parameters were chosen as $Q = 16$, $W = 57$, $N = 69$, $K = 8$, $M = 1$. The channel has been modeled as type Vehicular/A with a mobile velocity of 120 km/h. See the text for a discussion of the results.

9. 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.

Both the block-FFTs and the inversion of the diagonalized system matrix consist of independent subproblems that can be solved in parallel. Therefore the presented algorithm does not only reduce the computational complexity, but also exhibits inherent parallelism that can be exploited to decrease the computation time.

References

[1] M. Haardt and W. Mohr, "The complete solution for third generation mobile communications: Two modes on air - one winning strategy," in *Proc. IEEE Int. Conf. on Third Generation Wireless Communications*, June 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 Diver-

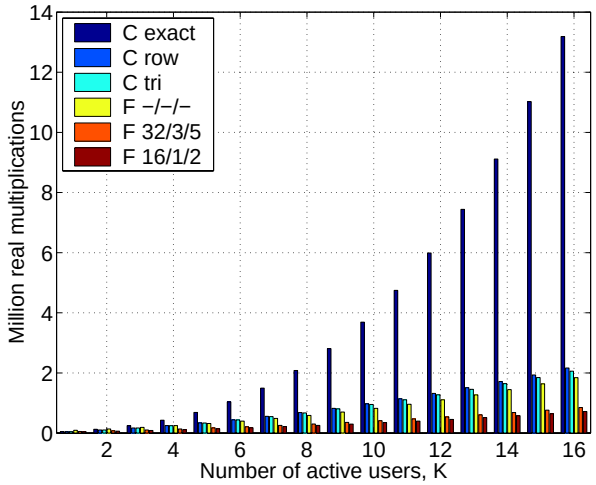


Figure 10: Computational complexity of the joint detection algorithms, as a function of the number of active codes, K. $Q = 16$, $W = 57$, $N = 69$, $M = 1$.

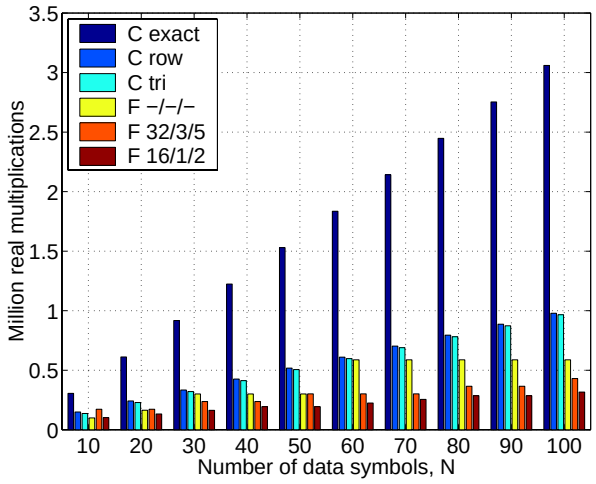


Figure 11: Computational complexity of the joint detection algorithms, as a function of the number of data symbols, N. $Q = 16$, $W = 57$, $K = 8$, $M = 1$.

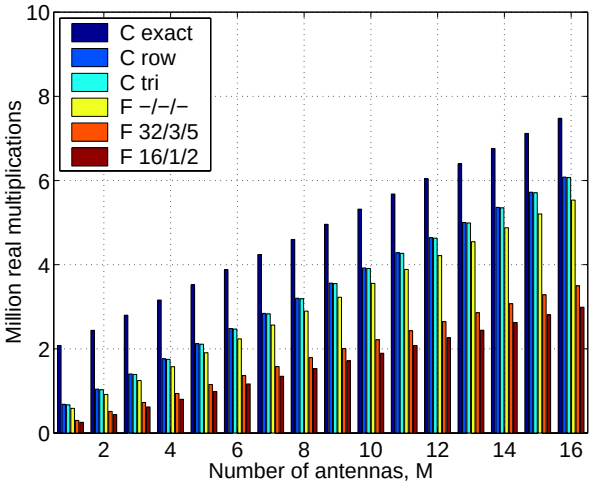


Figure 12: Computational complexity of the joint detection algorithms, as a function of the number of antennas, M. $Q = 16$, $W = 57$, $N = 69$, $K = 8$.

sity," *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.

[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] 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.

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

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

[9] C.F. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM Publications, Philadelphia, PA, 1992.

[10] 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.

[11] P. Jung, J. J. Blanz, and P. W. Baier, "Coherent receiver antenna diversity for CDMA mobile radio systems using joint detection," in *Proc. 4th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, Yokohama, Japan, Sept. 1993, pp. 488–492.

[12] J. 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 J. Select. Areas Commun.*, vol. 12, pp. 568–579, May 1994.

[13] B. Steiner and P.W. Baier, "Low Cost Channel Estimation in the Uplink Receiver of CDMA Mobile Radio Systems," *FREQUENZ*, vol. 47, pp. 292–298, 1993.

[14] L. Eldén and E. Sjöström, "Fast computation of the principle singular vectors of Toeplitz matrices arising in exponential data modelling," *Signal Processing*, vol. 50, no. 1-2, pp. 151–164, April 1996.

[15] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, third edition, 1996.