A Cordic Based Equalizer For Multiuser Detection In WCDMA Systems

Benjamin Heyne, Jürgen Götze
University of Dortmund, Information Processing Lab
Otto-Hahn-Str. 4, 44227 Dortmund, Germany
E-Mail: benjamin.heyne@uni-dortmund.de
Phone: +49 231 755 7017, FAX: +49 231 755 3251

Abstract— **Future terminals for CDMA will have to employ multiuser detection to implement high data rate modes such as HSDPA in the UTRA/3GPP standard.**

Therefore efficient and flexible detection algorithms are needed. In [1] we have already shown an approach of such an equalizer for single user detection. The principle algorithm of this equalizer has now been extended to a multiuser detector, which can make use of the same Cordic based platform as the original equalizer.

The paper shows that our approach has got a significant performance increase in comparison to a standard Rake based equalizer, whereas the computational complexity remains roughly the same.

I. Introduction

Third Generation (3G) terminals using Wideband Code Division Multiple Access (WCDMA) will soon have to implement multiuser detection schemes to support the growing demand of high data rate services like video on demand, web browsing or file transfers.

In UTRA/3GPP for example, this mode is called High Speed Downlink Packet Access (HSDPA) [2]. One logical data channel is spread over several physical channel, separated by individual spreading codes. Additionally higher order modulation (16-QAM) is used for the single channel.

Currently most of the WCDMA terminals are using Rake [3][4] receivers for symbol detection. But due to the Multiple Access Interference (MAI) in CDMA systems used to full capacity, this type of receiver becomes unusable.

Instead other detection methods like Parallel Interference Cancellation [5] have to be used. This paper presents another alternative, a Cordic based linear least squares multiuser equalizer/detector which comprises descrambling and despreading.

The original algorithm presented in [1] has been a single user detector. We now present an extension of that work to a multiuser detector to implement modes like HSDPA in UTRA/3GPP, or to enhance the performance of the detector in the single user modes.

The paper is organized as follows. At first the system model used for the algorithm is described in Section II. Then we will present the algorithm, the solution approach and the proposed implementation in Sections III-IV. In Section V, we will show some simulation results followed by the conclusions in Section VI.

II. System Model

Consider a CDMA downlink where all u users share the same channel. Thus the received signal also contains the signal of the other users. The system model used is shown in Figure 1.

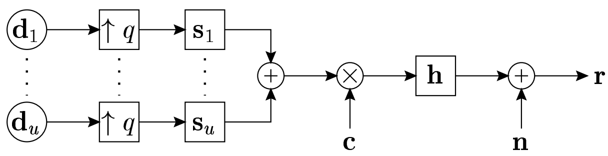


Fig. 1. Block diagram of the system model.

The incoming m complex data symbols of user i, collected in the vector $\mathbf{d}_i$, are first upsampled by the spreading factor q, so that one symbol now consists of q chips. Each upsampled symbol is now convolved with an OVSF code [6] contained in the vector $\mathbf{s}_i$ of length q. Finally, the summed data streams are scrambled with the complex data sequence in vector $\mathbf{c}$ which is repeated for every data frame (38400 chips in UTRA/FDD).

The received chips are now obtained by propagating the signal through a channel, which is characterized by its complex valued channel impulse response vector $\mathbf{h}$ of length h_l, and adding an AWGN component $\mathbf{n}$.

Therefore the received data vector $\mathbf{r}$ is given by

$$\mathbf{r} = \mathbf{n} + \mathbf{HC}\sum_{i=1}^{u}\mathbf{S}_i\mathbf{d}_i \tag{1}$$

where $\mathbf{H}$ is a convolutional matrix describing the time variant complex channel, $\mathbf{C}$ is a complex valued diagonal matrix containing the scrambling code $\mathbf{c}$ on its main diagonal and $\mathbf{S}_i$ is a block Toeplitz spreading matrix. In this matrix each block is one column wide and contains the OVSF code for the i-th data stream.

For the proposed algorithm it is assumed that the received signal $\mathbf{r}$ has already passed the chip matched filter and has been sampled at chip rate. We also assume that the channel impulse response $\mathbf{h}$ (or the strongest taps of it) is known, as the channel estimation is not part of this paper.

III. Multiuser Extension

The derivation of the single user detector/equalizer from this model is described in [1]. For the multiuser detection we are interested in the first j users and will treat the other users as an additional noise component, so that $\mathbf{r}$ is changed to

$$\mathbf{r} = \mathbf{n}' + \mathbf{HC}\sum_{i=1}^{j}\mathbf{S}_i\mathbf{d}_i \tag{2}$$

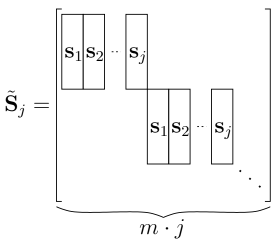
with

$$\mathbf{n}' = \mathbf{n} + \mathbf{HC}\sum_{i=j+1}^{u}\mathbf{S}_i\mathbf{d}_i\,.$$

To describe $\mathbf{r}$ as a simple matrix-vector multiplication, we create a new symbol vector $\tilde{\mathbf{d}}_j$ which contains the time interleaved data symbols $\mathbf{d}_i$ of the first j users

$$\tilde{\mathbf{d}}_j = [d_{11}\ \ d_{12}\ \ \cdots\ \ d_{1j}\ \ d_{21}\ \ d_{22}\ \ \cdots\ \ d_{2j}\ \ \cdots\ \ d_{mj}]^T\,,$$

and a new spreading matrix $\tilde{\mathbf{S}}_j$ which contains the j corresponding spreading codes:



Now the vector $\mathbf{r}$ can be rewritten as:

$$\mathbf{r} = \mathbf{n}' + \mathbf{HC}\tilde{\mathbf{S}}_j\tilde{\mathbf{d}}_j \tag{3}$$

The structures and dimensions of the matrices for one data frame are clarified in Figure 2. The column vectors h_i of $\mathbf{H}$ are assumed to be constant for at least one symbol interval. The number of symbols m can be calculated from the spreading factor q and the length of the scrambling code c_l.

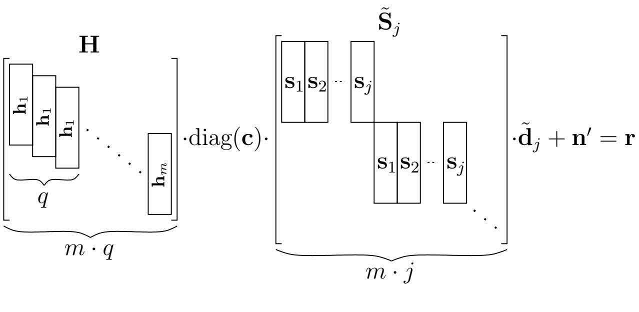


Fig. 2. Structures of the matrices used for the calculation.

When we have a close look at the structure of the matrices involved in the computation as shown in Figure 2, we will notice that there are several characteristic properties that can be exploited to simplify the calculation of the data symbols. In our approach the $\mathbf{H}$, $\mathbf{C}$ and $\tilde{\mathbf{S}}_j$ matrices are multiplied to get the system matrix $\tilde{\mathbf{K}}_j$ of width $m \cdot j$:

$$\tilde{\mathbf{K}}_j = \mathbf{HC}\tilde{\mathbf{S}}_j\,. \tag{4}$$

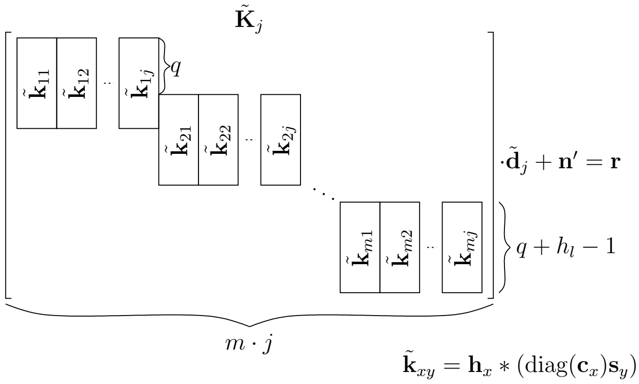


Fig. 3. Resulting structure of the system matrix $\tilde{\mathbf{K}}_j$. The $*$-operator indicates a convolution.

This matrix will be used to calculate the desired data symbols. The structure of $\tilde{\mathbf{K}}_j$ is shown in Figure 3.

The nonzero column vectors are calculated by convolving the channel impulse response with the scrambled spreading code. $\mathbf{c}_x$ denotes the x-th code block of length q in $\mathbf{c}$. As the scrambled spreading code just has got $(\pm 1 \pm \mathrm{i})$ entries the system matrix can be build without using multiplications.

It is obvious that $\tilde{\mathbf{K}}_j$ has got a very sparse structure which can be exploited as described in Section IV, to reduce the computational effort to solve the linear system.

IV. Implementation

The detection of the estimated data symbols $\tilde{\mathbf{d}}'_j$ can now be performed by solving the overdetermined linear system

$$\tilde{\mathbf{K}}_j\tilde{\mathbf{d}}'_j = \mathbf{r}\,. \tag{5}$$

The equation is solved in the least squares sense by a QR-decomposition which can be implemented efficiently on a processor array [7][8] using Cordic processor elements (PE) performing Givens transformations. As the structure of $\tilde{\mathbf{K}}_j$ is known, a direct approach is used for the calculation of the QR decomposition. A new system matrix is build from $\tilde{\mathbf{K}}_j$ and $\mathbf{r}$. Then the required Givens transformations are applied only to the nonzero elements of $\tilde{\mathbf{k}}_{xy}$ in $\tilde{\mathbf{K}}_j$. This approach is shown in Figure 4.

In each step one vector $\tilde{\mathbf{k}}_{xy}$ is annihilated. For each annihilation of one element in $\mathbf{k}_{xy}$ it is necessary to recompute two rows of the matrix composed of $\tilde{\mathbf{K}}_j$ and $\mathbf{r}$. The last step shows the matrix $\mathbf{R}$ and the vector $\mathbf{r}'$ which are used to perform the back-substitution.

The decomposition of $\tilde{\mathbf{K}}_j = \mathbf{QR}$ leads to a staircase like structure of the $\mathbf{R}$ matrix as shown in Figure 5.

If w is equal to the maximum number of overlapping vectors $\tilde{\mathbf{k}}_{xy}$ in one row of $\tilde{\mathbf{K}}_j$, then p is the maximum number of overlapping blocks $p = \frac{w}{j}$ in a row. The step size is equal to j. This will help to reduce the number of operations necessary to compute the back-substitution.

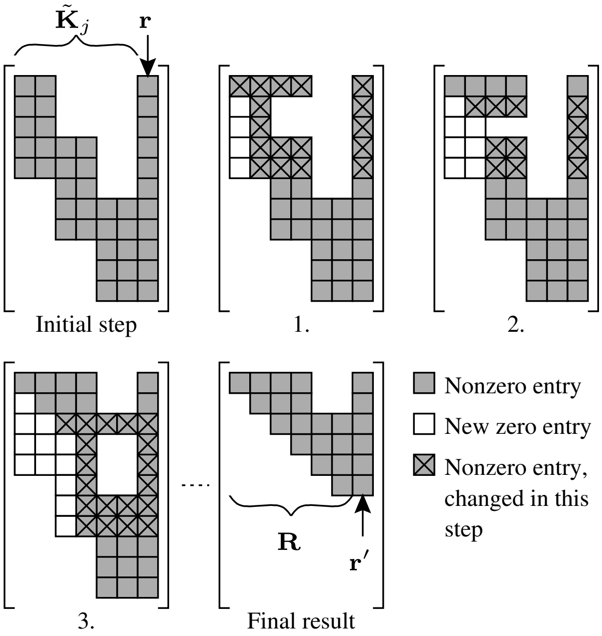


Fig. 4. Computations for the direct solution of the linear system.

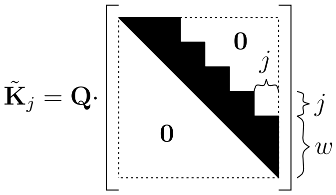


Fig. 5. Resulting structure of the $\mathbf{R}$ matrix.

A. Subdividing the Calculation

It is obvious that $\tilde{\mathbf{K}}_j$ can grow to a very large matrix. A whole data frame of $m \cdot j = 2400 \cdot 16 = 38400$ symbols at spreading factor $q = 16$ and a channel of length $h_l = 10$ would require a $\tilde{\mathbf{K}}_j$ matrix of size 38409×38400.

To overcome this problem the system $\tilde{\mathbf{K}}_j \tilde{\mathbf{d}}'_j = \mathbf{r}$ is subdivided into overlapping subsystems of manageable size as shown in Figure 6 and described in [9].

The linear system is subdivided into blocks of size $m_b \cdot j$. Using this method it is possible to solve the complete system without the need to store the whole matrix which would also involve large latency and memory needs. Overlapping of the blocks is necessary, as the independent calculation of the subproblems leads to higher symbol errors at the block edges as shown in Figure 7. This method involves a certain amount of computational overhead as the grey blocks have to be calculated twice. The overhead can be reduced by choosing larger block sizes m_b.

For good results the overlapping factor should be chosen at least as high as the block overlapping factor p of $\tilde{\mathbf{K}}_j$. As the columns can be calculated independently, it is also possible to update the channel $\mathbf{h}$ during the calculation of the blocks.

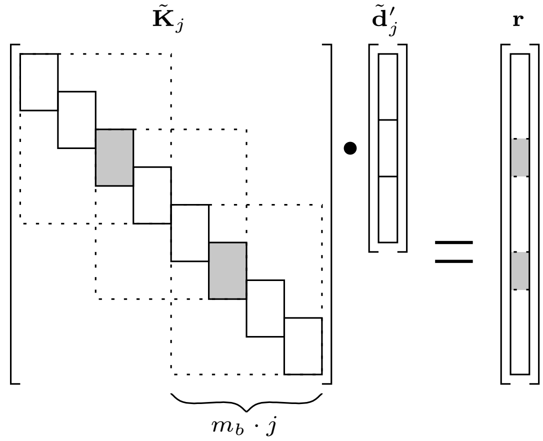


Fig. 6. Overlapping for easier calculation.

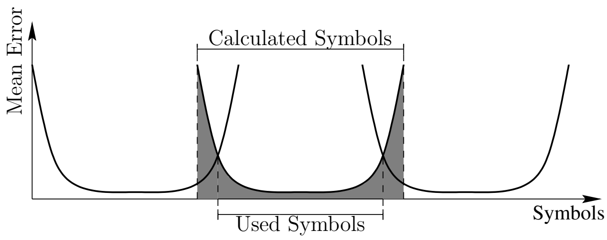


Fig. 7. Expected error function.

B. Computational Complexity

At first we will determine the number of Cordic operations necessary to perform the QR decomposition. The Cordic based decomposition needs three Cordic operations for the calculation of the complex angle at the beginning of each row, and four for each complex rotation operation per column. Therefore the number of operations is equal to:

$$\mathrm{OP_{QR}} = \sum_{i=1}^{m_b j} \sum_{k=k_{\mathrm{Start}}}^{q+h_l+\lfloor (i-1)/j \rfloor q-1} 3+4 + \sum_{l=i+1}^{m_b j} 4 \cdot (\tilde{\mathbf{K}}_j(k,l) \neq 0 \vee \tilde{\mathbf{K}}_j(i,l) \neq 0)$$
$$k_{\mathrm{Start}} = \begin{cases} i+1 & \text{for} \quad \lfloor (i-1)/j \rfloor q \leq i \\ \lfloor (i-1)/j \rfloor q + 1 & \text{else} \end{cases} . \tag{6}$$

$\tilde{\mathbf{K}}_j(x,y)$ denotes the corresponding matrix element in $\tilde{\mathbf{K}}_j$. The back-substitution, when performed on the Cordic processor array, will need:

$$\mathrm{OP_{Back}} = 4 + \sum_{i=1}^{m_b j-1} 4 + \sum_{k=i+1}^{m_b j} 9 \cdot (\mathbf{R}(i,k) \neq 0) \tag{7}$$

Cordic operations. Nine operations are needed for a complex MAC operation, and four for the closing division.

Finally, to compute the system matrix itself,

$$\mathrm{OP_{SM}} = m_b j h_l q \quad \text{with} \quad q \geq h_l \tag{8}$$

complex additions have to be performed.

If the overlap method is used for a data frame with $q = 16$, $m_b = 8$, $p = 2$, $j = 16$ and $m = 2400$ about 400 blocks have to be calculated for one frame. It is also assumed that the channel length h_l is 10.

In this case the decomposition/back-substitution for each block needs ≈ 179000 (real valued) Cordic operations and ≈ 20500 complex additions for the creation of the system matrix. Therefore the detection of a whole data-frame of $m \cdot j = 38400$ symbols uses

$$\approx 1865 \quad \frac{\text{Cordic Operations}}{\text{Symbol}}$$

and

$$\approx 214 \quad \frac{\text{Complex Additions}}{\text{Symbol}}.$$

Note that there is no further descrambling/despreading necessary. Furthermore the Cordic based QR decomposition can make 100% use of two parallel Cordics.

The complexity comparison of our proposed algorithm to other implementations is based on the numbers given in [10] for Rake and PIC based receivers. An equivalent of three array multipliers for one Cordic operation is used to include some overhead. Therefore a Cordic operation would be equal to three operations, and a complex addition equals two operations. Then complexity of our approach for this example is on the same order than the conventional Rake receiver as shown in Table I.

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISON.

	Rake	**Cordic based LS**
$\frac{\textbf{Operations}}{\textbf{Symbol}}$	4352	6023

Of course this is only a rough estimate of the computational complexity. But it shows that it is about the same as for the conventional Rake receiver, while the performance is significantly increased as shown in Section V.

C. Hardware Platform

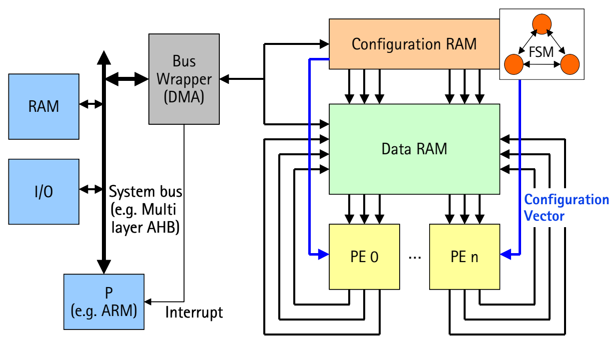


Fig. 8. The RACE coprocessor.

The equalizer is implemented on the programmable hardware accelerator (called RACE) shown in Figure 8. It can be described as an algorithm specific instruction set processor (ASIP) with a limited instruction set that is optimized for different classes of algorithms. The accelerator contains several processing elements (PE) in parallel that perform the computations, a data RAM to store values and a configuration RAM in conjunction with a finite state machine (FSM) to control the data flow [11].

In our case the class of algorithms is composed of matrix based algorithms that can be implemented by enhanced Cordic operations. For this purpose the accelerator contains Cordic processing elements which are based on simple shift-add operations, but there are other PE types such as e.g. MACs that can be used as well. The operations that can be performed, and which are used to implement the QR decomposition of the system matrix, are given in Table II.

TABLE II
BASIC OPERATIONS OF THE CORDIC PE.

Mode	Operation
Ortogonal Rotation	$x_{out} = x_{in}\cos(\phi_z) + y_{in}\sin(\phi_z))$ $y_{out} = -x_{in}\sin(\phi_z) + y_{in}\cos(\phi_z))$ $z_{out} = z_{in}$
Ortogonal Vector	$x_{out} = \sqrt{x_{in}^2 + y_{in}^2}$ $y_{out} = 0$ $z_{out} = \arctan_2(x_{in}, y_{in})$
Linear Rotation	$x_{out} = x_{in}$ $y_{out} = -x_{in}z_{in} + y_{in}$ $z_{out} = z_{in}$
Linear Vector	$x_{out} = x_{in}$ $y_{out} = 0$ $z_{out} = y_{in}/x_{in}$

For example in the "Orthogonal Rotation" mode the Cordic rotates a two dimensional input vector $\mathbf{a} = (x, y)$ by an angle ϕ_z.

V. SIMULATIONS

Figure 9 shows the frame error rate for a 16-QAM based system with $q = 16$, $j = 16$, $h_l = 10$, $m_b = 8$ and $p = 2$. The channel is assumed to be constant throughout the simulation and contains four, randomly distributed, strong taps. Hence the Rake receiver is using four fingers.

The Figure shows that the Rake has got no chance to detect the symbols in this case, and that the LS approach has got a large performance gain for rising SNR values. Performance and complexity comparisons to PIC receiver architectures are under examination. It is expected that the performance is comparable to our LS approach, but the complexity of a PIC based receiver is much higher.

VI. CONCLUSIONS

Our single-user detector has been extended to a solely Cordic based multiuser detector for WCDMA systems. The computational complexity of the presented algorithm can be compared to that of a Rake receiver, whereas the simulations show a significant performance gain over the Rake.

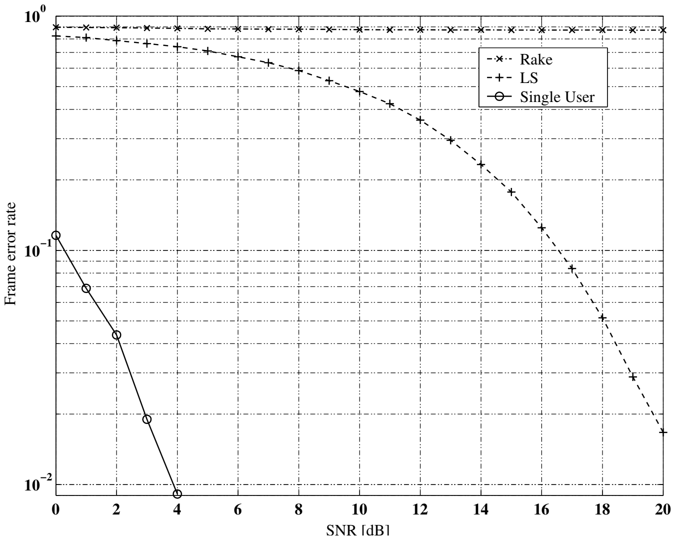


Fig. 9. Error rate for 16-QAM modulation.

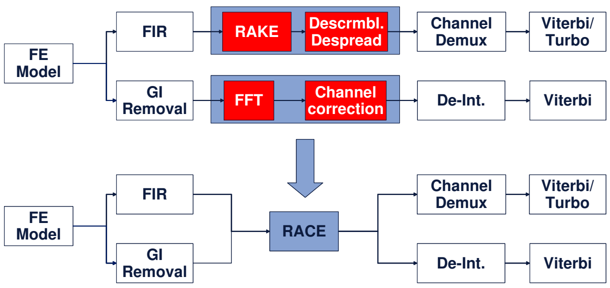


Fig. 10. Software defined baseband processing for WLAN and UTRA using the RACE coprocessor.

Future work will include the comparison to other detectors like PIC receivers and the effects of using time variant channels as described in Section IV-A.

The algorithm to calculate the FFT on a Cordic based architecture has been shown in [12]. Hence a CDMA equalizer and parts of an OFDM receiver can now be build upon solely Cordics. This can be used to implement a reconfigurable (software defined) architecture for multi standard terminal digital basebands, replacing dedicated hardware by using the RACE accelerator (Figure 10).

References

[1] B. Heyne, M. Otte, and J. Goetze, "A Performance Adjustable and Reconfigurable CDMA Receiver Concept for UMTS-FDD," in *14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC2003)*, Beijing, China, September 2003.

[2] 3GPP, "TR25.858 V5.0.0 - High Speed Downlink Packet Access: Physical Layer Aspects," Technical specification group radio access network, 3rd Generation Partnership Project (3GPP), 2002.

[3] J. G. Proakis, *Digital Communications*, McGraw-Hill, New york, 2nd edition, 1989.

[4] Keshab K. Parhi and Takao Nishitani, *Digital Signal Processing for Multimedia Systems*, Marcel-Dekker, New York, 1999.

[5] M. Varasani and B. Aazhang, "Multistage Detection in Asynchronous Code-Division Multiple-Access Communications," *IEEE Transactions on Commun.*, vol. 38, pp. 509–519, 1990.

[6] 3GPP, "TS25.213 V5.0.0 - Spreading and Modulation (FDD)," Technical specification group radio access network, 3rd Generation Partnership Project (3GPP), 2002.

[7] M. Otte, J. Goetze, and M. Buecker, "Matrix Based Signal Processing on a Reconfigurable Hardware Accelerator," in *10th Digital Signal Processing Workshop*, Pine Moutain, Georgia, USA, October 2002.

[8] S. Haykin, *Adaptive Filter Theory*, Prentice Hall (3rd edition), 1995.

[9] M. Vollmer, M. Haardt, and J. Götze, "Comparative Study of Joint-Detection Techniques for TD-CDMA Based Mobile Radio Systems," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1461–1475, August 2001.

[10] A. Nahler, R. Irmer, and G. Fettweis, "Reduced And Differential Parallel Interference Cancellation For CDMA Systems," *IEEE J. on Select. Areas in Commun.*, vol. 20 (2), pp. 237–247, 2002.

[11] B. Oelkrug, M. Buecker, D. Uffmann, A. Droege, J. Brakensiek, and M. Darianian, "Programmable Hardware Accelerator for Universal Telecommunication Applications," in *2nd Workshop on Software Radios*, Karlsruhe, Germany, 2002.

[12] B. Heyne and J. Goetze, "A Pure Cordic Based FFT For Reconfigurable Digital Signal Processing," in *12th European Signal Processing Conference (EUSIPCO2004)*, Vienna, Austria, September 2004.