Efficient CORDIC Based Equalizer For STTD Encoded MIMO CDMA 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*— Next generation mobile terminals will have to support multiuser detection and more than one antenna (Multiple Output Multiple Input - MIMO) to implement the latest high data rate modes of UTRA/3GPP such as HSDPA.**

Hence, very efficient and flexible detection algorithms are needed. In [1] we have already shown an approach of such an equalizer for multi user detection. In this paper we will present a multi user MIMO linear least squares based equalizer for Space-Time Transmit Diversity (STTD) encoded data, which can make use of the same CORDIC based platform as the multi user equalizer.

The paper shows that our approach has got a significant performance increase while maintaining computational complexity in comparison to a standard Rake based equalizer.

I. Introduction

The latest Third Generation (3G) terminals using Wideband Code Division Multiple Access (WCDMA) are implementing 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.

To further improve data rates or error resistance MIMO (Multiple Output Multiple Input) modes have been incorporated in the standard [3]. Using multiple antennas, a further gain in bandwith becomes possible.

On the other hand, the hardware becomes much more complex. Two antennas need two HF-frontends, and two baseband processing units in the worst case. This paper presents the extension of the CORDIC based linear least squares multiuser equalizer/detector presented in [1] to support the STTD (Space-Time Transmit Diversity) based MIMO mode of [3] using two antennas. This extension is still based on CORDIC computations only. Therefore, the complete computation can still take place on the architecture already defined in [4].

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 data streams share the same channel. Thus, the received signal also contains the signal of all the other users. The system model used is shown in Figure 1.

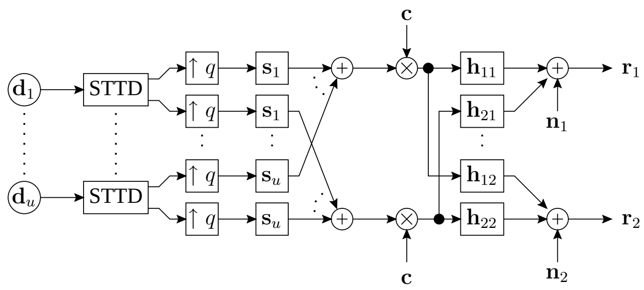
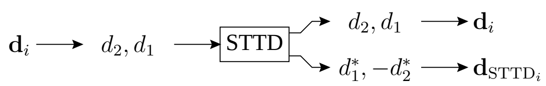


Fig. 1. Block diagram of the system model.

The incoming m complex data symbols of user i, collected in the vector $\mathbf{d}_i$, are firstly encoded using the STTD scheme (a * denotes the complex conjugate - QPSK modulation):



Next, all symbols are upsampled by the spreading factor q, so that one symbol now consists of q chips. All upsampled symbols are now convolved with OVSF codes [5] contained in the vectors $\mathbf{s}_i$ of length q. Each STTD encoded data stream $\mathbf{d}_i$ is convolved with the corresponding vector $\mathbf{s}_i$.

Finally, the summed top and bottom 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) to obtain the transmit data for both antennas.

The received chips are obtained by propagating the signal through four channels, which are characterized by their complex valued channel impulse response vectors $\mathbf{h_{kl}}$ of length h_l, and added AWGN components $\mathbf{n}_p$ at each antenna p.

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

$$\mathbf{r_1} = \mathbf{n_1} + \mathbf{H}_{11}\mathbf{C}\sum_{i=1}^{u}\mathbf{S}_i\mathbf{d}_i + \mathbf{H}_{21}\mathbf{C}\sum_{i=1}^{u}\mathbf{S}_i\mathbf{d}_{\mathrm{STTD}_i} \quad (1)$$

and

$$\mathbf{r_2} = \mathbf{n_2} + \mathbf{H}_{12}\mathbf{C}\sum_{i=1}^{u}\mathbf{S}_i\mathbf{d}_i + \mathbf{H}_{22}\mathbf{C}\sum_{i=1}^{u}\mathbf{S}_i\mathbf{d}_{\mathrm{STTD}_i} \quad (2)$$

where $\mathbf{H}_{kl}$ are the a convolutional matrices describing the time variant complex channels, $\mathbf{C}$ is a complex valued diagonal matrix containing the scrambling code $\mathbf{c}$ on its main diagonal and $\mathbf{S}_i$ are the block Toeplitz spreading matrices. In these matrices 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 signals $\mathbf{r}_p$ have already passed the chip matched filter and have been sampled at chip rate. We also assume that the channel impulse responses $\mathbf{h}_{kl}$ are known, as the channel estimation is not part of this paper.

III. Algorithm

The CORDIC based multi user detector/equalizer is described in [1]. Now we will describe the derivation of the CORDIC based MIMO STTD equalization algorithm.

We are interested in the first j users and will treat the other users as an additional noise component. Also, we assume one estimated channel per antenna. Hence, the received signal $\mathbf{r}_p$ is changed to

$$\mathbf{r}_p = \mathbf{n}'_p + \mathbf{H}_p\mathbf{C}\sum_{i=1}^{j}\mathbf{S}_i\mathbf{d}_i^p \quad (3)$$

with

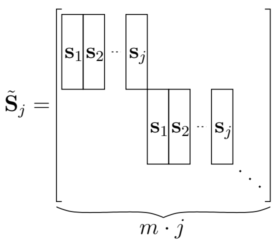
$$\mathbf{n}'_p = \mathbf{n}_p + \mathbf{H}_p\mathbf{C}\sum_{i=j+1}^{u}\mathbf{S}_i\mathbf{d}_i^p\,.$$

Thus, we would have to solve p equation systems (one for each antenna). To be able to apply a block processing scheme to equalize the data, all equation systems are combined into one. We start by deriving the systems at each antenna, and then interleave them into one.

At first, new symbol vectors $\tilde{\mathbf{d}}_j^p$ are created, which contain the time interleaved data symbols $\mathbf{d}_i^p$ of the first j users

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

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



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

$$\mathbf{r}_p = \mathbf{n}' + \mathbf{H}_p\mathbf{C}\tilde{\mathbf{S}}_j\tilde{\mathbf{d}}_j^p \quad (4)$$

The structures and dimensions of the matrices for one data frame at each antenna are clarified in Figure 2. The column vectors h_i^p of $\mathbf{H}_p$ 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.

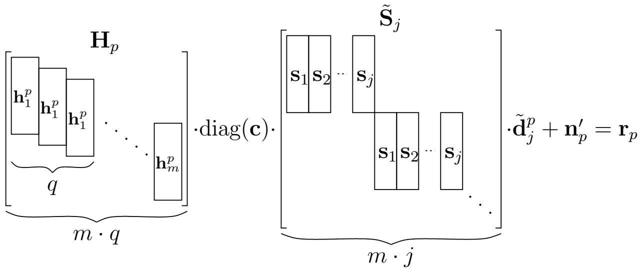


Fig. 2. Structures of the matrices used for the calculation.

Several characteristic properties of the matrices involved in the computation as shown in Figure 2 can be exploited to simplify the calculation of the data symbols. In our approach the $\mathbf{H}_p$, $\mathbf{C}$ and $\tilde{\mathbf{S}}_j$ matrices are multiplied to get the system matrices $\mathbf{K}_j^p$ of width $m \cdot j$ for each antenna:

$$\mathbf{K}_j^p = \mathbf{H}_p\mathbf{C}\tilde{\mathbf{S}}_j\,. \quad (5)$$

The structure of $\mathbf{K}_j$ is shown in Figure 3. The nonzero column

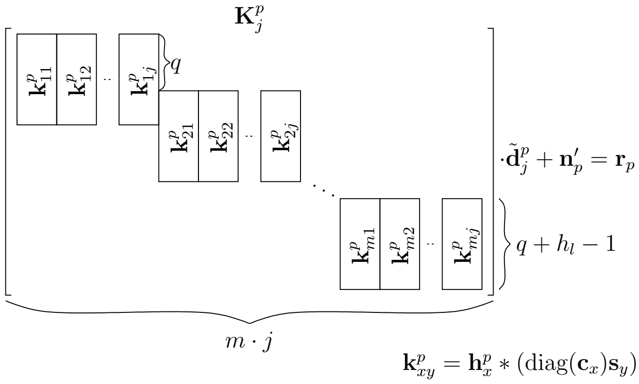


Fig. 3. Resulting structure of the system matrix $\tilde{\mathbf{K}}_j$. The $*$-operator indicates a convolution.

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.

In order to incorporate all p equation systems into one, all $\mathbf{r}_p$ and $\mathbf{K}_j^p$ are multiplexed into one received chip vector

$$\mathbf{r} = [r_1^1 \;\; r_1^2 \;\; \cdots \;\; r_1^p \;\; r_2^1 \;\; r_2^2 \;\; \cdots \;\; r_{m\cdot j}^p]^T\,,$$

and one system matrix $\tilde{\mathbf{K}}_j$ with each column vector

$$\tilde{\mathbf{k}}_{xy} = [k_{1xy}^1 \;\; k_{1xy}^2 \;\; \cdots \;\; k_{1xy}^p \;\; k_{2xy}^1 \;\; k_{2xy}^2 \;\; \cdots \;\; k_{(q+h_l-1)xy}^p]^T\,,$$

corresponding to row interleaving all $\mathbf{K}_j^p$ matrices. 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

$$\tilde{\mathbf{K}}_j \tilde{\mathbf{d}}'_j = \mathbf{r} \tag{6}$$

with

$$\tilde{\mathbf{d}}'_j = [d_{11} \;\; d_{12} \;\; \cdots \;\; d_{1j} \;\; d_{21} \;\; d_{22} \;\; \cdots \;\; d_{2j} \;\; \cdots \;\; d_{mj}]^T .$$

The equation is solved in the least squares sense by a QR-decomposition which can be implemented efficiently on a processor array [6][7] 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.

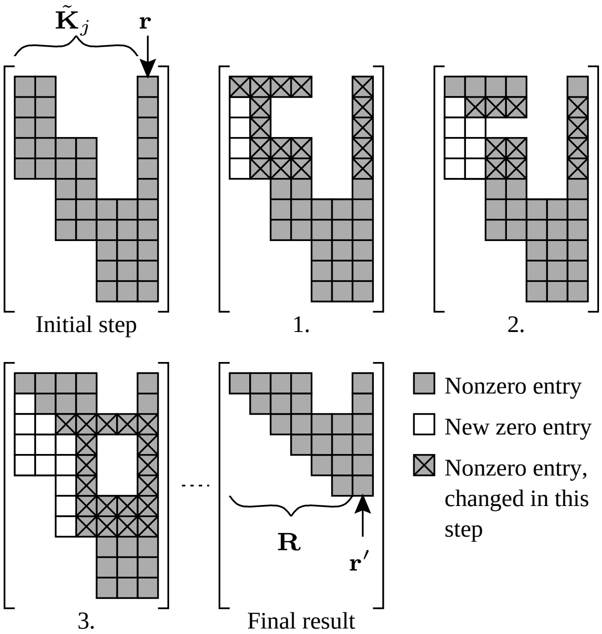


Fig. 4. Computations for the direct solution of the linear system.

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 b is the maximum number of overlapping blocks $b = \frac{w}{j}$ in a row. The step size is equal to

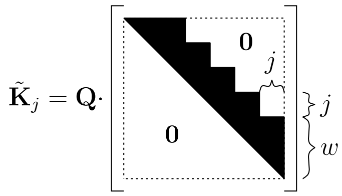


Fig. 5. Resulting structure of the $\mathbf{R}$ matrix.

j. This will help to reduce the number of operations necessary to compute the back-substitution.

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 $q = 16$, $p = 2$ antennas and a channel of length $h_l = 10$ would require a $\tilde{\mathbf{K}}_j$ matrix of size 76818×38400.

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 [8].

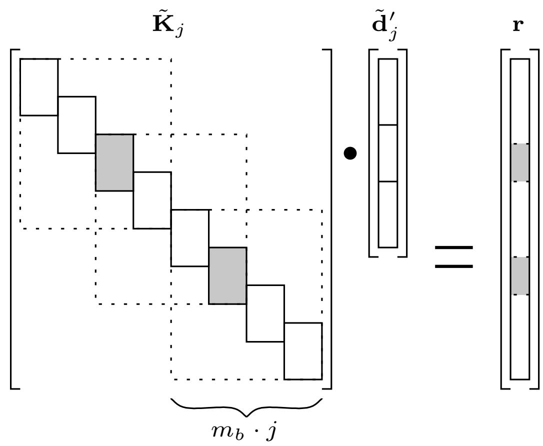


Fig. 6. Overlapping for easier calculation.

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 b of $\tilde{\mathbf{K}}_j$, due to the implied influence on neighbouring blocks. As the columns can be calculated independently, it is also possible to update the channel $\mathbf{h}$ during the calculation of the blocks.

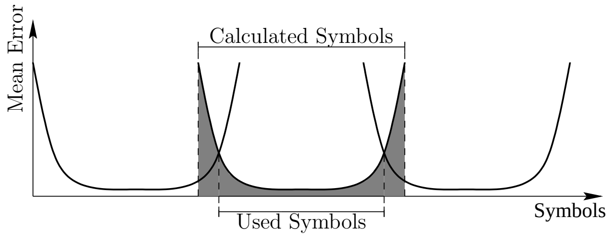


Fig. 7. Expected error function.

B. STTD Decoding

The final step is the decoding of the STTD code described in Section II. This can also be performed by just using CORDICs in orthogonal rotation mode (Table I) as shown in Figure 8.

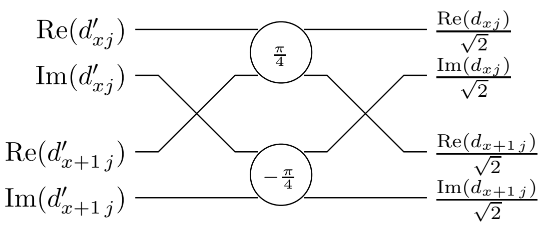


Fig. 8. CORDIC based STTD decoding.

The decoding must be performed on every two consecutive symbols of each spreading code for all symbols to obtain the decoded data symbols $\tilde{\mathbf{d}}_j$. As the real and imaginary parts of the symbols can be processed in parallel, this stage just adds one CORDIC operation in time per symbol using two parallel CORDICs. If the overlap method is used to calculate the symbols, the STTD decoding has to be performed just on the used symbols of one block.

C. 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:

$$\begin{aligned} \text{OP}_{\text{QR}} = \sum_{i=1}^{m_b j} & \sum_{k=k_{\text{Start}}}^{p\cdot(q+h_l+\lfloor (i-1)/j \rfloor q-1)} 3+4 \\ & + \sum_{l=i+1}^{m_b j} 4\cdot(\tilde{\mathbf{K}}_j(k,l)\neq 0 \vee \tilde{\mathbf{K}}_j(i,l)\neq 0) \\ k_{\text{Start}} = & \begin{cases} i+1 & \text{for} \quad \lfloor (i-1)/j \rfloor q \leq i \\ p\cdot(\lfloor (i-1)/j \rfloor q+1) & \text{else} \end{cases} . \end{aligned} \tag{7}$$

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

$$\text{OP}_{\text{Back}} = 4 + \sum_{i=1}^{m_b j-1} 4 + \sum_{k=i+1}^{m_b j} 9\cdot(\mathbf{R}(i,k)\neq 0) \tag{8}$$

CORDIC operations. Nine operations are needed for a complex MAC operation, and four for the closing division. STTD decoding needs

$$\text{OP}_{\text{STTD}} = 2\cdot\lfloor m_b/2 \rfloor j \tag{9}$$

operations. Finally, to compute the system matrix itself,

$$\text{OP}_{\text{SM}} = p m_b j h_l q \quad \text{with} \quad q \geq h_l \tag{10}$$

complex additions have to be performed.

If the overlap method is used for a data frame with $q = 16$, $m_b = 8$, $b = 2$, $p = 2$, $j = 16$ 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/STTD decoding for each block needs 431000 $\approx$ (real valued) CORDIC operations and $\approx$ 41000 complex additions for the creation of the system matrix. Therefore the detection of a whole data-frame of $m\cdot j = 38400$ symbols uses

$$\approx 4490 \quad \frac{\text{CORDIC Operations}}{\text{Symbol}}$$

and

$$\approx 428 \quad \frac{\text{Complex Additions}}{\text{Symbol}} .$$

Note that these numbers inlude descrambling, despreading and STTD decoding. 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 [9] for Rake and PIC based receivers. Usually an equivalent of two to three array multipliers for one CORDIC operation is used as a complexity estimate. Therefore a CORDIC operation would be equal to 2-3 operations, and a complex addition equals two operations. Hence, our CORDIC based equalizer would use between 9836 and 14326 operations per symbol. A Rake based implementation ($p = 2$ Rakes) would need about 9000 operations per symbol.

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

D. Hardware Platform

The equalizer is implemented on the programmable hardware accelerator (called RACE) shown in Figure 9. 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

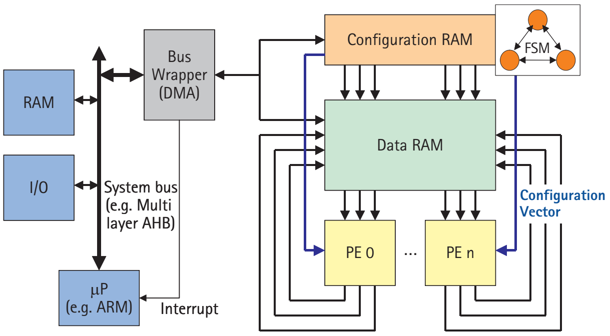


Fig. 9. The RACE coprocessor.

RAM in conjunction with a finite state machine (FSM) to control the data flow [4].

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 I.

TABLE I
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 $\mathbf{a} = [x\, y]^T$ by an angle ϕ_z.

V. SIMULATIONS

Figure 10 shows the bit error rate for a 16-QAM based system with $q = 16$, $j = 16$, $h_l = 10$, $p = 2$, $m_b = 8$ and $b = 2$. The channels are assumed to be constant throughout the simulation and contain four, randomly distributed, strong taps. The estimated channel at each antenna in MIMO mode is calculated as the mean of the transmission channels. 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, which is further improved by adding the second antenna.

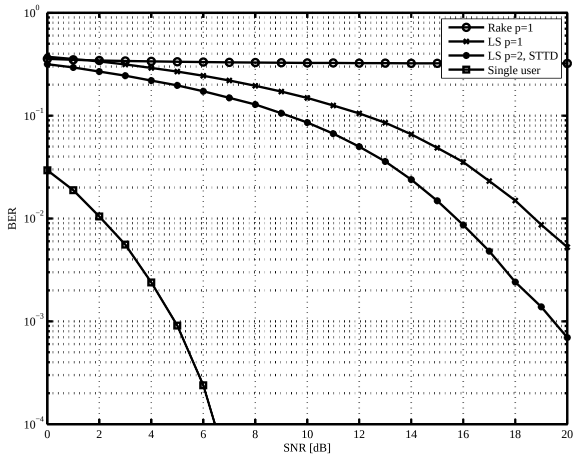


Fig. 10. Bit Error Rate for 16-QAM modulation.

VI. CONCLUSIONS

We have presented an algorithm for a low complexity, solely CORDIC based, linear least squares equalizer including despreading and descrambling for MIMO WCDMA systems. If necessary, CORDIC based STTD decoding may also be performed. 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.

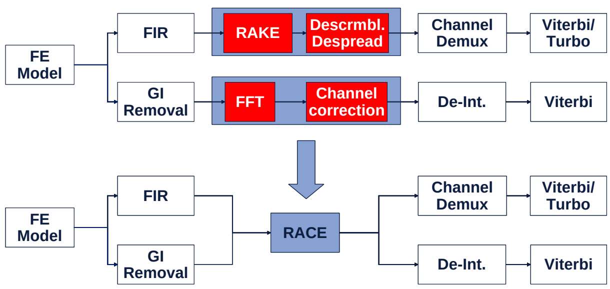


Fig. 11. Software defined baseband processing for WLAN and UTRA using the RACE coprocessor.

Future work will include the comparison to other detectors like frequency domain equalizers 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 [10]. Hence a MIMO WCDMA equalizer including STTD decoding 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 11).

References

[1] B. Heyne and J. Goetze, "A Cordic Based Equalizer For Multiuser Detection in WCDMA Systems," in *IEEE Workshop on Signal Processing Systems (SiPS2005)*, Athens, Greece, November 2005.

[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] 3GPP, "TS25.211 V7.1.0 - Physical channels and mapping of transport channels onto physical channels (FDD)," Technical specification group radio access network, 3rd Generation Partnership Project (3GPP), 2007.

[4] 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.

[5] 3GPP, "TS25.213 V5.0.0 - Spreading and Modulation (FDD)," Technical specification group radio access network, 3rd Generation Partnership Project (3GPP), 2002.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.