A CORDIC BASED RECONFIGURABLE CDMA EQUALIZER FOR LONG CODES AND ITS COMPARISON TO THE RAKE RECEIVER

Benjamin Heyne, Juergen Goetze
University of Dortmund, Information Processing Lab
Otto-Hahn-Str. 4, 44227 Dortmund, Germany
E-Mail: benjamin.heyne@uni-dortmund.de, Phone: +49-231-7557017.

ABSTRACT

This paper presents the implementation of a solely Cordic based linear equalizer for CDMA (including descrambling and dispreading) on a software defined architecture. The performance of this approach is compared to the Rake receiver. We formulate the different detection concepts in a common matrix notation and discuss their performance in an UTRA-FDD environment. Simulations confirm the presented results.

1. INTRODUCTION

A lot of modern signal processing applications require such a high computational power that only ASICs can fulfill the technical demands. Unfortunately, ASICs are inflexible, costly (development and debugging) and only economical for mass-products. As a consequence, system designers are striving to replace specialized hardware solutions with software based solutions as developments in the field of software radio demonstrate. Due to the fact that even the most commonly used programmable devices, i.e. DSPs, often lack the required processing power, one tries to develop a solution that lays somewhere in between the two extrema *programmable signal processing* and *dedicated hardware*. The efforts in this area are summarized with the term *reconfigurable computing*.

Within the framework of the MoReTeX[1] project this approach is used to create a common software defined baseband implementation for UTRA-FDD and WLAN as shown in Figure 1. The most computational intensive tasks are performed on a dedicated hardware accelerator called RACE using Cordic processing elements. The implementation of the FFT on a Cordic based architecture has been shown in [11]. This paper presents the implementation of a Cordic based linear equalizer, including descrambling and despreading, on the same architecture. The original Rake [1-4] receiver can also be described in our matrix-vector notation, such that it can be compared to our approach.

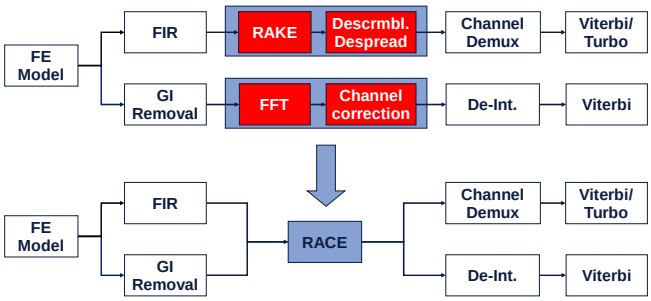

Figure 1: Software defined baseband processing for WLAN and UTRA using the RACE accelerator

As the equalizer and the FFT can now be build upon solely Cordics, we have derived a reconfigurable (software defined) architecture for a mobile multi-standard terminal and the main processing blocks of the WLAN and UMTS baseband can be replaced by this programmable architecture.

The paper is organized as follows. At first, the RACE accelerator architecture is described in Section 1. Then the system model, followed by a derivation of the Cordic based algorithm is presented. The comparison to the Rake and a channel inversion approach is shown in Sections 2-5. Then an overview of the implementation is given in Section 6. In Section 7 we show the simulation results in an UTRA-FDD environment, followed by the conclusions in Section 8.

2. ACCELERATOR

The equalizer is implemented on the programmable hardware accelerator (RACE [7]) shown in Figure 2. The

[1] This project is partly funded by the German Ministry of Education and Research (BMBF) in the Mobile, Reconfigurable Multi-standard Terminal for UMTS and WLAN (MoReTeX) project, which is a part of the umbrella Software Defined Radio Based Architecture Studies for Re-configurable Mobile Communication Systems (RMS), No. 01 BU171.

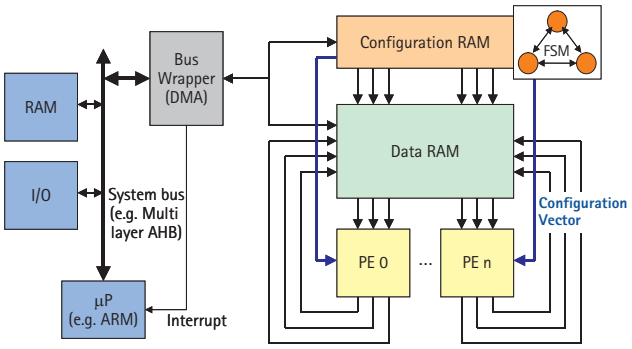

Figure 2: RACE architectural overview

RACE can be described as an algorithm specific instruction set processor (ASIP) with a limited instruction set that is optimized for different classes of algorithms. In this case the class is composed of matrix based algorithms that can be performed by enhanced CORDIC operations.

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.

Here, the RACE is embedded in a processor environment where it is connected to the system bus via a bus wrapper, which has direct memory access (DMA) capability and thereby controls the dataflow into and out of the RACE. The processor itself is freed from all data moving tasks and is just informed by an interrupt when the results of an operation are available.

3. SYSTEM MODEL

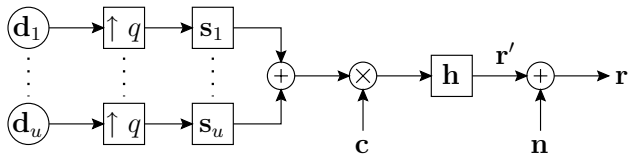

Figure 3: System model

Consider a CDMA downlink where all u users share the same channel and the first user is the desired one. Thus the received signal also contains the signal of the other users. The system model used is shown in Figure 3.

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 [8] contained in the vector $\mathbf{s}_i$ of length q. Finally, the summed data streams are scrambled with the complex data sequence in vector $\boldsymbol{c}$ which is repeated for every data frame and has got 38400 elements (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

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

where $\mathbf{H}$ is a convolution matrix describing the per-symbol 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 $\mathbf{h}$ is known, as the channel estimation is not a part of this paper. As we are just interested in the symbols of the first user, we will treat the other users as an additional noise component. Therefore:

$$\mathbf{r} = \mathbf{n}' + \mathbf{HCS}_1\mathbf{d}_1 \quad \text{with} \quad \mathbf{n}' = \mathbf{n} + \mathbf{HC}\sum_{i=2}^{u}\mathbf{S}_i\mathbf{d}_i \qquad (2)$$

The structures and dimensions of these matrices for one data frame are clarified in Figure 4. The column vectors $\mathbf{h}_i$ of $\mathbf{H}$ are assumed to be constant for at least one symbol interval.

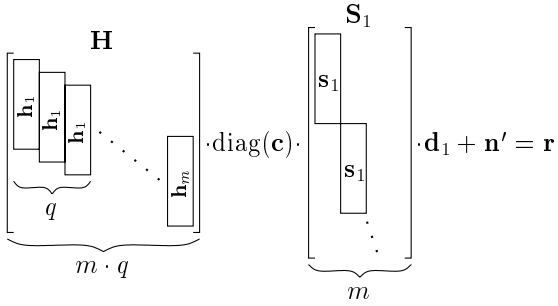

Figure 4: Structures of the involved matrices

The number of symbols m can be calculated from the spreading factor q and the length of the scrambling code c_l.

4. THE RAKE RECEIVER

Consider a four finger Rake receiver as shown in Figure 5. Each finger x is fixed to one tap h_x of the current channel impulse response $\mathbf{h}$ at chip-time offset τ_x ($h(\tau_x)=h_x$) The received signal is therefore passed through the four filters described by the convolution matrices $\mathbf{H}_1$-$\mathbf{H}_4$.

Note that the coefficients and time offsets may change after one symbol period.

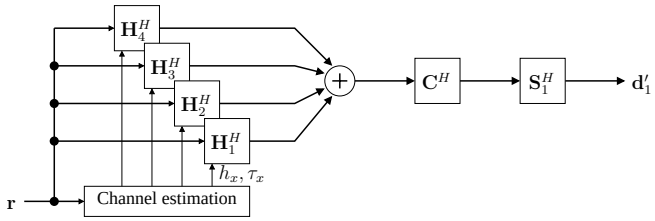

Figure 7: Rake receiver principle

Basically these matrices are diagonal matrices having one channel coefficient h_x on the τ_x-th diagonal, corresponding to the tap delay. The results of these filters are accumulated, descrambled by $\mathbf{C}^H$ and finally despread using $\mathbf{S}_1^H$ to obtain the users symbols $\mathbf{d}_1$. The filtering of $\mathbf{r}$ by the four filters $\mathbf{H}_1$-$\mathbf{H}_4$ can also be described by one single filter having the convolution matrix $\hat{\mathbf{H}} = \mathbf{H}_1 + \mathbf{H}_2 + \mathbf{H}_3 + \mathbf{H}_4$. The difference between $\hat{\mathbf{H}}$ and $\mathbf{H}$ is just that the channel coefficients not used for the Rake receiver are set to zero in $\hat{\mathbf{H}}$ (Figure 6). Hence, the estimated symbols for the Rake receiver are derived by:

$$\mathbf{d}_{\text{Rake}} = \mathbf{S}_1^H \mathbf{C}^H \hat{\mathbf{H}}^H \mathbf{r} \tag{3}$$

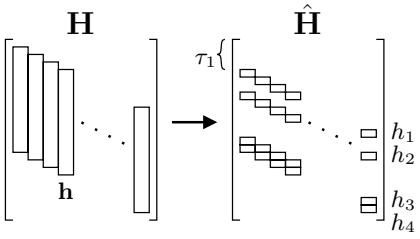

Figure 6: The resulting channel matrix (assuming a constant channel) for the Rake receiver.

5. THE CORDIC BASED LINEAR EQUALIZER

When we have a close look at the structure of the matrices involved in the computation as shown in Figure 4, 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 **H**, **C** and $\mathbf{S}_1$ matrices are multiplied to get the system matrix $\mathbf{K} = \mathbf{HCS}_1$ of width *m*. This matrix will be used to calculate the desired data symbols. The structure of **K** is shown in Figure 7. The nonzero column vectors are calculated by convolving the channel impulse response with the scrambled spreading code. As the scrambled spreading code just has got $\pm 1 \pm j$ entries the system matrix can be build without using multiplications.

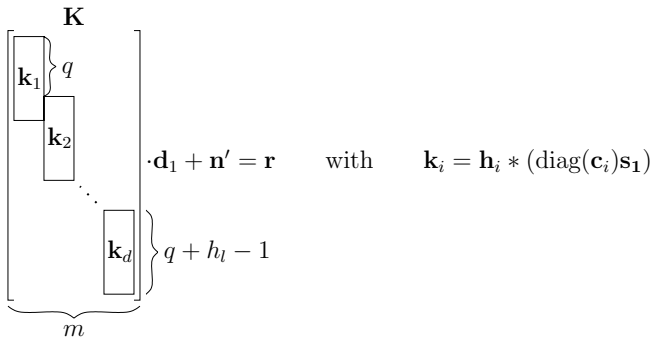

Figure 5: Structure of the system matrix K.

It is obvious that **K** has got a very sparse structure which can be exploited as described in Section 6, to reduce the computational efford to solve the linear system.

5.1. Symbol estimation

By now the detection of the data symbols can be simplified to the form $\mathbf{K}\hat{\mathbf{d}}_1 = \mathbf{r}$. This overdetermined linear system of equations can now easily be solved in the least squares (LS) sense by a QR-decomposition which can be implemented efficiently on a systolic processor array [5][6] using Cordic processor elements (PE) to perform Givens-transformations. Therefore the estimated data symbols are obtained as:

$$\mathbf{d}_{\text{LS}} = \mathbf{K}^{\#}\mathbf{r} = (\mathbf{K}^H\mathbf{K})^{-1}\mathbf{K}^H\mathbf{r} = (\mathbf{K}^H\mathbf{K})^{-1}\mathbf{S}_1^H\mathbf{C}^H\mathbf{H}^H\mathbf{r} \tag{4}$$

Our LS approach, when used as an alternative to the Rake receiver, estimates the symbols by using $\hat{\mathbf{K}} = \hat{\mathbf{H}}\mathbf{CS}_1$ instead of **K**. Therefore it only uses the same channel taps as the Rake receiver. Then:

$$\hat{\mathbf{d}}_{\text{Rake}} = \hat{\mathbf{K}}^H\mathbf{r} \quad \text{and} \quad \hat{\mathbf{d}}_{\text{LS}} = \hat{\mathbf{K}}^{\#}\mathbf{r} = (\hat{\mathbf{K}}^H\hat{\mathbf{K}})^{-1}\hat{\mathbf{K}}^H\mathbf{r} \tag{5}$$

5.2. Comparison to the Rake receiver

Obviously the only difference between the Rake receiver and our linear equalizer is the term $(\hat{\mathbf{K}}^H\hat{\mathbf{K}})^{-1}$. If a one tap channel is assumed, this factor is reduced to just a scaling of the symbols (the column vectors in $\hat{\mathbf{K}}$ do not overlap) and the performance is equal to the Rake (Figure 8). When the length of **h** grows and the $\hat{\mathbf{k}}_i$ in $\hat{\mathbf{K}}$ start overlapping, the secondary diagonals in $\hat{\mathbf{K}}^H\hat{\mathbf{K}}$ will get occupied, too. The number of occupied secondary diagonals in $\hat{\mathbf{K}}^H\hat{\mathbf{K}}$ is equal to the maximum number of $\hat{\mathbf{k}}_i$ overlapping in one row of $\hat{\mathbf{K}}$. Also, the values on the main secondary diagonals become very small (one or two decimal powers) if the spreading factor is large compared to the length of the channel. Therefore, $\hat{\mathbf{K}}^H\hat{\mathbf{K}}$ can be approximated by just calculating the main diagonal values in these cases.

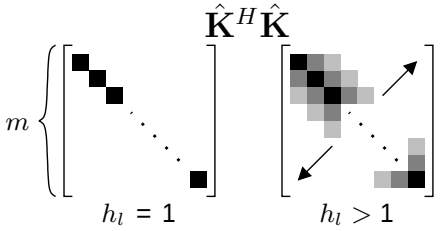

Figure 8: Structure of $\hat{\mathbf{K}}^H\hat{\mathbf{K}}$ depending on the channel length.

So one could say the LS approach is canceling the inter symbol interference, which the Rake does not. Even a Rake utilizing the whole channel impulse response will never achieve the performance of our approach.

5.3. Comparison to LS channel equalization only

Several other approaches exists which equalize the channel by a least squares algorithm, but do the despreading and descrambling separately. In this case the estimated symbols are derived as:

$$\begin{aligned}\mathbf{d}_{\text{ChLS}} &= \mathbf{S}_1^H\mathbf{C}^H\mathbf{H}^{\#}\mathbf{r} = \mathbf{S}_1^H\mathbf{C}^H(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{r}\\ \hat{\mathbf{d}}_{\text{ChLS}} &= \mathbf{S}_1^H\mathbf{C}^H\hat{\mathbf{H}}^{\#}\mathbf{r} = \mathbf{S}_1^H\mathbf{C}^H(\hat{\mathbf{H}}^H\hat{\mathbf{H}})^{-1}\hat{\mathbf{H}}^H\mathbf{r}\end{aligned} \quad (6)$$

In the noiseless case this approach gives the best performance. The Rake receiver has got the worst performance then, as it is the matched filter approach to the channel only LS. Our approach's performance lies in-between, as it is a Rake like matched filter but with an approximate elimination of ISI.

5.4. Interpreting the simulation results

If we have a look at the simulation results in Figure 12 of Section 7, it can be seen that the Rake and our linear equalizer approach have got a better performance for low SNRs (which is crucial in mobile applications) compared to the channel only LS, whereas the channel equalizer gets better for high SNRs. To understand this behavior, we have a look at the noise enhancement of the three different approaches. As $\mathbf{r}$ can be expressed as $\mathbf{n}'+\mathbf{r}_1$ for $\mathbf{r}_1 = \mathbf{HCS}_1\mathbf{d}_1$ the noise dependent terms are:

$$\begin{aligned}\mathbf{n}_{\text{Rake}} &= \mathbf{S}_1^H\mathbf{C}^H\mathbf{H}^H\mathbf{n}'\\ \mathbf{n}_{\text{LS}} &= \mathbf{K}^{\#}\mathbf{n}' = (\mathbf{K}^H\mathbf{K})^{-1}\mathbf{S}_1^H\mathbf{C}^H\mathbf{H}^H\mathbf{n}'\\ \mathbf{n}_{\text{ChLS}} &= \mathbf{S}_1^H\mathbf{C}^H\mathbf{H}^{\#}\mathbf{n}' = \mathbf{S}_1^H\mathbf{C}^H(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{n}'\end{aligned} \quad (7)$$

Note that $\mathbf{n}'$ is not white anymore – It also contains the other user's data. However, it is easy to see that the noise enhancement of the Rake and the linear equalizer performs similar. Again the LS approach has got a small advantage because of the ISI correction. In case of the channel LS, the effects of the colored noise become obvious as shown in the simulation results (Section 7).

6. IMPLEMENTATION OF THE EQUALIZER

The over determined linear system $\mathbf{d}_{\text{LS}}=\mathbf{K}^{\#}\mathbf{r}$ is solved in the least squares sense by a QR-decomposition which can be implemented efficiently using Cordic processor elements (PE) performing Givens-transformations.

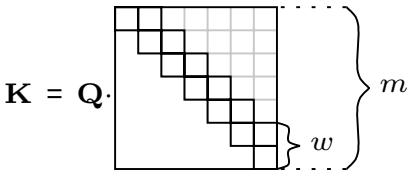

Figure 9: Structure of the R matrix

Usually this is done using a systolic processor array[5], but due to the special, sparse structure of $\mathbf{K}$ a systolic array would perform a lot of unnecessary operations. Also, the algorithm does not need to compute the whole $\mathbf{R}$ matrix. As shown in Figure 9, the computation of $\mathbf{R}$ leads to a banded matrix of band width w which is equal to the degree of overlapping of the column vectors $\mathbf{k}_i$ in $\mathbf{K}$. Hence, just the first w diagonals of $\mathbf{R}$ have to be computed.

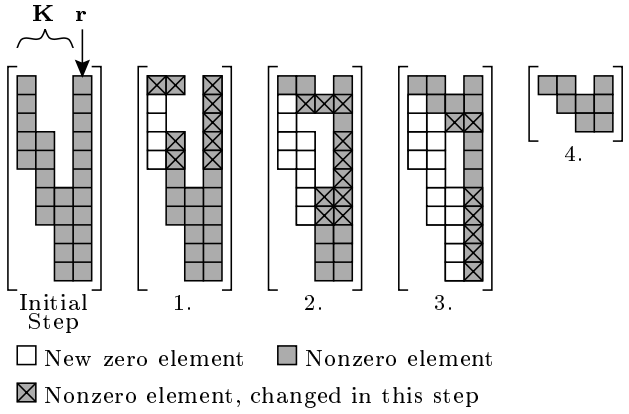

Figure 10: Algorithm for the direct calculation of R.

As the structure of $\mathbf{K}$ is known exactly, it is possible to apply the required Givens-Transformations only to the nonzero elements of $\mathbf{k}_i$ in $\mathbf{K}$. This approach is shown in Figure 10. In each step one vector $\mathbf{k}_i$ is annihilated. For each annihilation of one element in $\mathbf{k}_i$ it is necessary to recompute two rows of the matrix composed of $\mathbf{K}$ and $\mathbf{r}$. The last step shows the matrix that is used to perform the backsubstitution.

It is obvious that **K** can grow to a very large matrix. A whole data frame of 150 symbols at spreading factor q=256 and a channel of length h_l=40 would require a **K** matrix of size 38439×150. To overcome this problem the linear system is subdivided into overlapping subsystems of manageable size as described in [10]. A more detailed description of the implementation can be found in [12].

The algorithm leaves a lot of space for approximations to save computational resources. The number of nonzero channel taps taken into account, or the accuracy for the calculation of **R** are two examples. When implemented on the RACE accelerator this corresponds to just a change of the microcode. Currently the implementation of the equalizer on a RACE utilizing two parallel Cordic PEs needs about 11000 activations to calculate eight data symbols at h_l=10 and q=256.

7. SIMULATION RESULTS

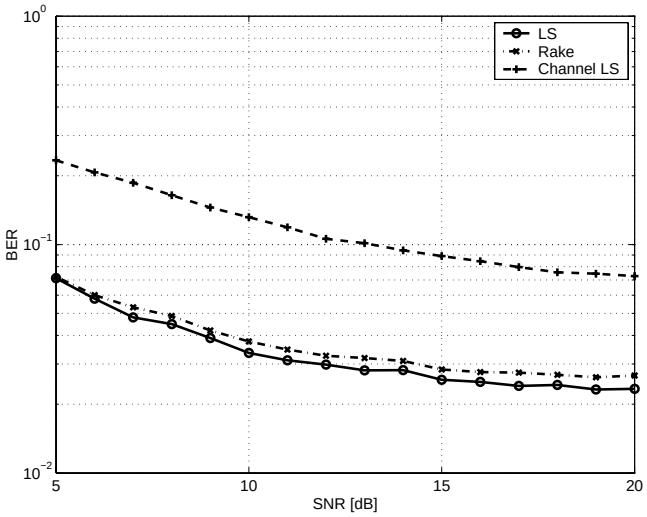

Figure 11: BER for Rake, LS and channel LS using $\hat{\mathbf{H}}$

For the comparison of the three different approaches to each other we have used a Matlab based simulation using a spreading factor of 16 and a channel using 10 taps with four strong taps. The system also contains 4 other users. The scrambling and channelization codes have been choosen according to the UTRA-FDD standards [8]. The channel was assumed to be known perfectly. To calculate the symbols the overlapping algorithm described in [12] has been used.

Figure 11 shows the uncoded BER for the three different approaches if $\hat{\mathbf{H}}$ is used as the channel convolution matrix for all equalizers. The results show that in this case the channel LS has got a disadvantage because of the bad condition of $\hat{\mathbf{H}}^{\#}$. In Figure 12 $\hat{\mathbf{H}}$ is still used for the Rake receiver, but the LS equalizers are using **H** now. In this case the channel LS gets better with a rising SNR, because the quality of $\mathbf{H}^{\#}$ is much better then.

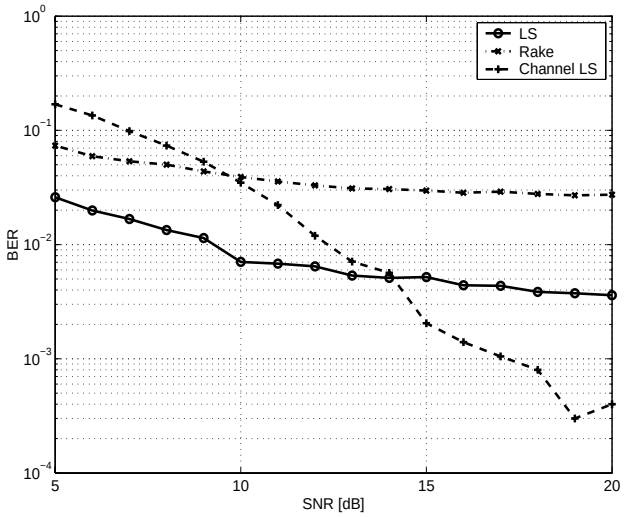

Figure 12: BER for Rake using $\hat{\mathbf{H}}$, LS and channel LS using **H**

Finally Figure 13 shows the coded BER of the Cordic based approach in an UTRA-FDD simulation. RACE was used to calculate the data symbols. The Rake is implemented with four fingers, and the LS equalizer is utilizing the complete channel impulse response as the impact on the computational complexity is relatively low [12]. As expected, the simulation confirms the Matlab based results.

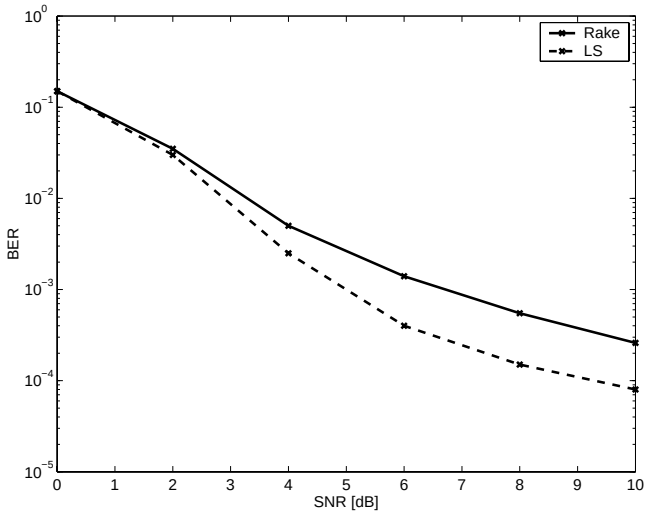

Figure 13: BER in UTRA-FDD simulation Rake vs. LS

8. CONCLUSIONS

Our Cordic based least squares approach to an UTRA-FDD equalizer has been compared to the Rake receiver and a channel inversion approach. We have derived a common matrix based notation for all of these approaches and compared the performance in an UTRA-FDD environment. It was shown that our LS approach has got better performance compared to the Rake in any case. Compared to the channel inversion approach our equalizer has got an advantage for low SNRs.

The Cordic based LS algorithm has been implemented on a programmable hardware accelerator and integrated into an UTRA-FDD baseband receiver simulation. The simulations are confirming our results. Hence, combined with the results from [11], we are now able to implement large parts of the UTRA and WLAN baseband on the Cordic based programmable hardware accelerator.

9. REFERENCES

[1] J. G. Proakis, *Digital Communications*, 2nd ed. New york: McGraw-Hill, 1989.

[2] K. K. Parhi and T. Nishitani, *Digital Signal Processing for Multimedia Systems*. New York: Marcel-Dekker, 1999.

[3] K. Kammeyer, *Nachrichtenübertragung*, 2nd ed. B.G. Teubner, 1996.

[4] H. Holma and A. Toskala, *WCDMA for UMTS*. New York: Wiley & Sons, 2001.

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

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

[7] B. Oelkrug, M. Bücker, D. Uffmann, A. Dröge, J. Brakensiek, and M. Darianian, "Programmable hardware accelerator for universal telecommunication applications," in *2nd Workshop on Software Radios*, Karlsruhe, Germany, 2002.

[8] "TS25.213 V5.0.0 - Spreading and Modulation (FDD)," 3rd Generation Partnership Project (3GPP)," Technical Specification Group Radio Access Network, 2002.

[9] R. Machauer, "Multicode-Detektion im UMTS," Ph.D. dissertation, University of Karlsruhe, INT, 2002.

[10] 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.

[11] B. Heyne, J. Götze: "A Pure Cordic Based FFT for Reconfigurable Digital Signal Processing". *12th European Signal Processing Conference (Eusipco2004)*, Vienna, Austria, September 2004.

[12] B. Heyne, M. Otte, J. Götze: "A Performance Adjustable and Reconfigurable CDMA Receiver Concept for UMTS-FDD". *14th IEEE International Symposium of Personal, Indoor and Mobile Radio Communications PIMRC2003*, Beijing, China, September 2003.