14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

MIMO SIGNAL PROCESSING ON A RECONFIGURABLE ARCHITECTURE

Klaus Hueske, Marius Otte, Jürgen Götze

Information Processing Lab, University of Dortmund
Otto-Hahn-Str. 4, 44221, Dortmund, Germany
Phone: + (49) 231 755 3036, Fax: + (49) 231 755 7019
E-Mail: klaus.hueske@udo.edu

Jörg Brakensiek

Nokia Research Center
Bochum, Germany
E-Mail: jorg.brakensiek@nokia.com

ABSTRACT

In this paper the implementation of multiple-input multiple-output (MIMO) signal processing on a reconfigurable hardware architecture is discussed. The implementation of MIMO systems is usually determined by the parameters of the application at hand, e.g. the number of sensor elements, the number of output signals or the required word length. Furthermore, there is also a flexibility in terms of the algorithms, which are used for computing the required task. We will present two different approaches for solving the linearly constrained MVDR (minimum variance distortionless response) beamforming problem. The two methods can be mapped on a reconfigurable hardware architecture. This architecture is described as a virtual systolic array, which consists of reconfigurable processor elements that can execute different transformation modes (linear, orthogonal). We will discuss the configuration in terms of change of parameters and change of algorithm, respectively. Furthermore, bit true simulations of the BER for the different approaches are presented for various word lengths. Finally, the trade-off between performance and reconfiguration effort is discussed.

1. INTRODUCTION

The growing demand for high rate wireless communication systems has drawn a great attention to MIMO communication techniques, in which multiple antennas are used for transmission and reception [1]. The increasing number of receiving antennas also leads to an increasing complexity of the signal processing.

Furthermore the quantity of supported transmission standards for wireless communications will grow. Like today with GSM, UMTS, WLAN and Bluetooth, all these standards have to be handled by one mobile device. To manage this, hardware seems to be too inflexible, so that a pure software baseband processing would be desirable (Software Defined Radio - SDR) [2],[3]. Unfortunately, today's processors cannot handle the full processing task in an energy efficient way, especially under the aspect of growing transfer rates.

This leads to the possible use of a reconfigurable hardware architecture [4], which lies somewhere in between SDR and dedicated hardware. This reconfigurable hardware works as an accelerator [5] for complex signal processing tasks and can be used with different standards (e.g. for solving constrained/unconstrained linear least squares problems, see Figure 1).

In this paper we will present a parallel implementation of a linearly constrained MVDR beamformer [6] on a reconfigurable architecture. Of course, the resulting architecture should support various parameter sets, as e.g. a different number of input signals or output signals. However, besides these system parameters, various algorithms used for the problem at hand can lead to different implementation requirements like e.g. the used word length. Here, we will discuss two approaches. Using the Schur complement in the first approach, the incorporation of the linear constraints into the minimization problem can be formulated as a partial Gaussian elimination process [7]. This requires a reconfiguration of the processor elements from orthogonal to linear mode. The second approach is based on a weighted embedding of the constraints into the minimization problem. The latter allows higher data throughput, since it could be executed without reconfiguration during the beamforming process. However, the first approach can be implemented with reduced word length.

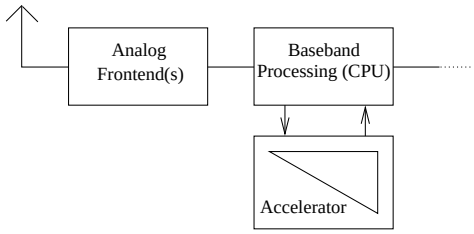

Figure 1: Software radio with accelerator

The paper is organized as follows: In section 2 we explain the underlying signal model and review the MVDR beamforming algorithm. In section 3 we present two different approaches for solving the constrained MVDR optimization problem. The used reconfigurable hardware architecture to implement the above approaches is described in section 4. Simulation results and a discussion concerning performance and configuration effort are given in sections 5 and 6.

2. SIGNAL MODEL

Consider a scenario with n omni-directional sensor elements located in a plane at positions $\mathbf{m}_i \in \mathbb{R}^2$. The antenna array receives a mixture of desired signals, undesired interferences from unknown direction and background noise that is assumed to be uniformly distributed over all directions. The scenario is depicted in Figure 2. Note that if $s_1(t)$ is the desired signal, the second signal $s_2(t)$ also represents an interferer with known direction of arrival, whereas $s_3(t)$ is an interferer from unknown direction. For simplicity, we use the far-field approximation and assume that $f_c \gg \beta$, with f_c the

This work was supported by NOKIA Research Center, Bochum, within the scope of the German government (BMBF) funded project "RMS Reconfigurable Mobile Communication System (No. 01BU171)".

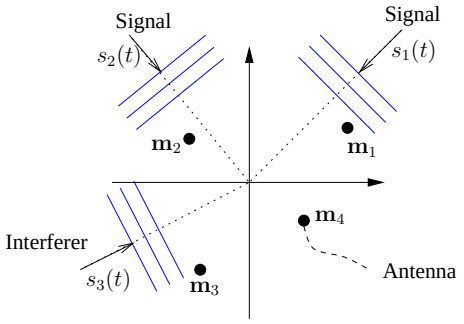

Figure 2: Scenario with four antennas

carrier frequency and β the bandwidth of the received signal. Therefore we can formulate the signal at the sensor element k in the baseband representation

$$x_k(t) = \tilde{x}(t)\exp(-j2\pi f_c \tau_k(\alpha)) + n_k(t), \tag{1}$$

where $\tilde{x}(t)$ is the complex baseband signal at a virtual reference sensor element placed in the origin of the antenna coordinate system and $\tau_k(\alpha)$ the time delay of a signal from direction α at sensor k relative to the reference sensor. The noise signal $n_k(t)$ should be a realization of a white Gaussian noise process (AWGN).
For receiving signals from p known directions using an array consisting of n antennas, we can formulate a more compact matrix based notation that is defined as follows:

$$\mathbf{C} = \begin{bmatrix} e^{j\phi_{1,1}} & e^{j\phi_{1,2}} & \dots & e^{j\phi_{1,n}} \\ e^{j\phi_{2,1}} & e^{j\phi_{2,2}} & \dots & e^{j\phi_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j\phi_{p,1}} & e^{j\phi_{p,2}} & \dots & e^{j\phi_{p,n}} \end{bmatrix} \text{ with } \phi_{\ell,k} = -2\pi f_c \tau_k(\alpha_l) \tag{2}$$

For a given array geometry and given directions of arrival α_l, the time delay $\tau_k(\alpha_l)$ can be easily calculated. The needed α_l have to be supplied by a Direction of Arrival (DOA) estimator [8], which is assumed to be optimal in this paper. The rows of $\mathbf{C}$ are also called steering vectors. By sampling the signal of each antenna element we get the discrete data matrix

$$\mathbf{X} = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_n(1) \\ x_1(2) & x_2(2) & \cdots & x_n(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(m) & x_2(m) & \cdots & x_n(m) \end{bmatrix}, \tag{3}$$

where m is the number of samples taken at each of the n antennas. Now, weighting the sensor outputs with complex factors w_i, the summation of these products results in a spatial filter, a so-called beamformer [9]. The output signal of this filter is given by

$$\begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix}_i = \mathbf{X} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}_i = \mathbf{X}\mathbf{w}_i \tag{4}$$

Every weight vector $\mathbf{w}_i$ corresponds to one desired output signal $\mathbf{e}_i$. For q signals we get the matrix notation

$$\mathbf{E} = [\mathbf{e}_1\ \mathbf{e}_2\ \cdots\ \mathbf{e}_q] = \mathbf{X}[\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_q] = \mathbf{X}\mathbf{W} \tag{5}$$

The directional characteristic of the array for one weight vector $\mathbf{w}$ is given by

$$\mathbf{d}_i = \begin{bmatrix} d_1 \\ \vdots \\ d_p \end{bmatrix}_i = \mathbf{C}\mathbf{w}_i \tag{6}$$

with $\mathbf{d}_i$ containing the amplitudes for all directions p. In order to determine the $\mathbf{w}_i$ for a desired direction, we have to define the values of $\mathbf{d}_i$. Generally values normalized to the set $\{0,1\}$ are used. For the elimination of a signal from a known direction we use 0, for a desired signal we use 1. The influence of all other unknown signals should be minimized, so we can formulate the beamforming problem as the constrained optimization problem

$$\min_{\mathbf{w}_i} \|\mathbf{e}_i = \mathbf{X}\mathbf{w}_i\|_2^2 \quad \text{for } i \in [1,q] \quad \text{and} \quad \mathbf{C}\mathbf{W} = \mathbf{D} \tag{7}$$

with $\mathbf{D} = [\mathbf{d}_1\,\mathbf{d}_2 \cdots \mathbf{d}_q]$ containing the constraints for all q desired signals.

3. ALGORITHMS

There are two different approaches to solve the above optimization problem.

3.1 Gauss-/Givens Transformation

One approach to solve the constrained optimization problem is to reformulate the problem as a least squares problem without constraints

$$\min_{\mathbf{w}_i} \|\mathbf{e}_i\|_2^2 \quad \text{with} \quad \begin{bmatrix} \mathbf{C} \\ \mathbf{X} \end{bmatrix} \mathbf{w}_i - \begin{bmatrix} \mathbf{d}_i \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{p,1} \\ \mathbf{e}_i \end{bmatrix} \tag{8}$$

With a partitioning of the matrices $\mathbf{C}$ and $\mathbf{X}$ we get

$$\begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \\ \mathbf{X}_1 & \mathbf{X}_2 \end{bmatrix} \begin{bmatrix} \mathbf{w}_{i1} \\ \mathbf{w}_{i2} \end{bmatrix} - \begin{bmatrix} \mathbf{d}_i \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{p,1} \\ \mathbf{e}_i \end{bmatrix} \text{ with } \begin{matrix} \mathbf{C}_1 \in \mathbb{F}^{p\times p} \\ \mathbf{X}_1 \in \mathbb{F}^{m\times p} \end{matrix} \tag{9}$$

Now the constraints equation can be written as

$$\mathbf{C}_1\mathbf{w}_{i1} + \mathbf{C}_2\mathbf{w}_{i2} = \mathbf{d}_i \tag{10}$$

Solving for $\mathbf{w}_{i1}$, we get

$$\mathbf{w}_{i1} = \mathbf{C}_1^{-1}(\mathbf{d}_i - \mathbf{C}_2\mathbf{w}_{i2}). \tag{11}$$

Thus,

$$\begin{aligned} \mathbf{X}\mathbf{w}_i &= \mathbf{X}_1\mathbf{w}_{i1} + \mathbf{X}_2\mathbf{w}_{i2} \\ &= (\mathbf{X}_2 - \mathbf{X}_1\mathbf{C}_1^{-1}\mathbf{C}_2)\mathbf{w}_{i2} - (-\mathbf{X}_1\mathbf{C}_1^{-1}\mathbf{d}_i) \end{aligned}$$

Therefore, with

$$\overline{\mathbf{X}}_2 = \mathbf{X}_2 - \mathbf{X}_1\mathbf{C}_1^{-1}\mathbf{C}_2 \tag{12}$$

and

$$\overline{\mathbf{y}}_i = -\mathbf{X}_1\mathbf{C}_1^{-1}\mathbf{d}_i \tag{13}$$

we have to solve the least squares problem

$$\min_{\mathbf{w}_{i2}} \|\overline{\mathbf{X}}_2\mathbf{w}_{i2} - \overline{\mathbf{y}}_i\|_2^2 \tag{14}$$

for all $i = 1, ..., q$ in order to get $\mathbf{W}_2$ and then calculate $\mathbf{W}_1$ from (11). In a further step we have to calculate the desired output signals by

$$\mathbf{E} = \mathbf{X}\mathbf{W}. \tag{15}$$

It is beneficial, particularly with regard to a parallel hardware implementation, to incorporate the steps involved in the solution of the constrained least squares problem (i. e. equations (11) till (15)) into one matrix triangularization process. This can be done by applying a sequence of Gaussian transformations $\mathbf{G}$ to equation 9. This leads to

$$\mathbf{G}\left(\begin{bmatrix}\mathbf{C}_1 & \mathbf{C}_2\\ \mathbf{X}_1 & \mathbf{X}_2\end{bmatrix}\mathbf{w}_i - \begin{bmatrix}\mathbf{d}_i\\ \mathbf{0}\end{bmatrix}\right) = \mathbf{G}\begin{bmatrix}\mathbf{0}_{p,1}\\ \mathbf{e}_i\end{bmatrix} \tag{16}$$

$$\Leftrightarrow \quad \begin{bmatrix}\mathbf{R}_1 & \mathbf{C}_2'\\ \mathbf{0}_{m,p} & \overline{\mathbf{X}}_2\end{bmatrix}\mathbf{w}_i - \begin{bmatrix}\mathbf{d}_i'\\ \overline{\mathbf{y}}_i\end{bmatrix} = \begin{bmatrix}\mathbf{0}_{p,1}\\ \mathbf{e}_i'\end{bmatrix}. \tag{17}$$

where $\mathbf{R}_1$ is an upper triangular matrix. The expressions $\overline{\mathbf{X}}_2$ and $\overline{\mathbf{y}}_i$ correspond to the Schur complement [10]. The application of $\mathbf{G}$ does not influence the result of the optimization problem, because $\mathbf{G}$ has the special structure

$$\mathbf{G} = \quad \begin{matrix}\\ p\end{matrix}\begin{matrix}\begin{matrix}1 & \\ & \ddots\end{matrix} & \\ & \begin{matrix}\ddots & 1\end{matrix}\end{matrix}\begin{matrix}\\ m\end{matrix} \tag{18}$$

The resulting least squares problem in the lower part of expression (17) can be solved by QR decomposition of $\overline{\mathbf{X}}_2$. This leads to

$$\begin{bmatrix}\mathbf{I}_p & \\ & \mathbf{Q}_m^H\end{bmatrix}\left(\begin{bmatrix}\mathbf{R}_1 & \mathbf{C}_2'\\ \mathbf{0}_{m,p} & \overline{\mathbf{X}}_2\end{bmatrix}\mathbf{w}_i - \begin{bmatrix}\mathbf{d}_i'\\ \overline{\mathbf{y}}_i\end{bmatrix}\right) = \begin{bmatrix}\mathbf{0}_{p,1}\\ \mathbf{Q}_m^H\mathbf{e}_i'\end{bmatrix} \tag{19}$$

$$\Leftrightarrow \begin{bmatrix}\mathbf{R}_1 & \mathbf{C}_2'\\ \mathbf{0}_{m,p} & \begin{bmatrix}\mathbf{R}_2\\ \mathbf{0}_{m-n+p,n-p}\end{bmatrix}\end{bmatrix}\mathbf{w}_i - \begin{bmatrix}\mathbf{d}_i'\\ \begin{bmatrix}\overline{\mathbf{y}}_{i1}\\ \overline{\mathbf{y}}_{i2}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\mathbf{0}_{p,1}\\ \mathbf{e}_i''\end{bmatrix}, \tag{20}$$

Note that the unitary transformation of $\mathbf{e}_i'$ does not change the solution of the equation as well, so we can solve the minimization problem by back substitution in the first n equations

$$\min_{\mathbf{w}_i}\|\mathbf{e}_i\|_2^2 = \min_{\mathbf{w}_i}\|\mathbf{e}_i''\|_2^2 \Leftrightarrow \begin{bmatrix}\mathbf{R}_1 & \mathbf{C}_2'\\ & \mathbf{R}_2\end{bmatrix}\mathbf{w}_i = \begin{bmatrix}\mathbf{d}_i'\\ \overline{\mathbf{y}}_{i1}'\end{bmatrix}. \tag{21}$$

This method can be implemented on a processor array, as described in section 4.

3.2 Weighting

An approximate solution $\hat{\mathbf{w}}_i$ of (7) can be calculated by embedding the weighted constraints into the minimization expression [11], like

$$\min_{\hat{\mathbf{w}}_i}\left\|\begin{bmatrix}\mathbf{X}\\ \xi\mathbf{C}\end{bmatrix}\hat{\mathbf{w}}_i - \begin{bmatrix}\mathbf{0}\\ \xi\mathbf{d}_i\end{bmatrix}\right\|_2^2 \tag{22}$$

with an amplification factor $\xi \in \mathbb{R}, \xi > 1$. With $\xi \to \infty$ the approximation $\hat{\mathbf{w}}_i$ converges against the optimum solution $\mathbf{w}_i$. This is a disadvantageous behavior for architectures with limited word length fixed point representations, because the risk of numerical overflows grows with ξ.
To obtain a good approximation with smaller ξ anyway, we can use the fact that the linear equations system is growing over time with a growing number of samples taken at the antennas. This means that we can interweave the matrix $[\xi\mathbf{C}|\xi\mathbf{d}_i]$ with the rows of $\mathbf{X}$ before passing the result to a least squares processor array as described in section 4. Instead of increasing the magnitude of ξ we can increase the density of the constraint equations.

4. ARCHITECTURE

The described approaches shall be implemented on a reconfigurable hardware architecture, which consists of a virtual systolic array similar to the one presented in [5]. Both algorithms are based on a triangularization process that can be expressed by transformation operations (linear, orthogonal). The first approach is based on a partial Gaussian elimination which can be expressed as a linear transformation. The subsequent QR decomposition can be done by unitary Givens transformations.
To get a more flexible homogeneous architecture, the PEs shall be able to perform both transformations. This leads to reconfigurable fixed-point CORDIC based processor elements (PEs), that are designed to operate in two modes, the orthogonal (Givens) and the linear (Gaussian) mode. Using approach 1 the processing starts with the calculation

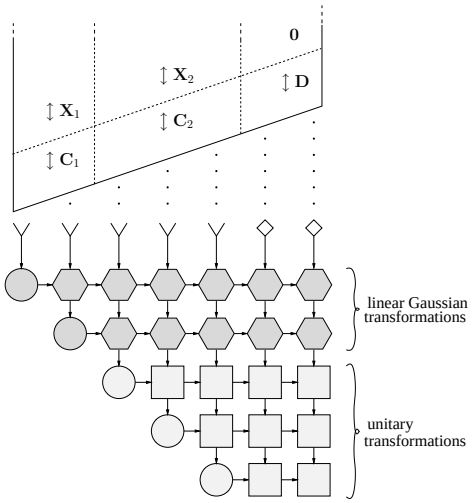

Figure 3: Processor array

of $\mathbf{R}_1$,$\mathbf{C}_2'$ and $\mathbf{d}_1' \cdots \mathbf{d}_2'$, which define the constraints. This is done by Gaussian transformations, that are realized by PEs in the linear mode. The particular implementation of the PEs makes it necessary to perform orthogonal operations before activating the linear mode for the Gaussian transformations. This is caused by the effort to construct PEs that are both simple and flexible.
The processing continues with the calculation of the QR decomposition in the last $n-p$ array rows, which is realized by unitary Givens rotations using the PEs in orthogonal mode. The resulting configuration is shown in Figure 3. Note that if the constraints are unchanged, the first p array rows stay constant. The modification of the constraints equations can be done by applying rows of the matrices $\mathbf{C}$ and $\mathbf{D}$ to the processor array without interrupting the data flow. However, when doing this we have to repeat the above initialization with a change between orthogonal and linear mode. Because

this operation causes some delay, the throughput of the processor array using this approach is decreased.
The triangularization process of method 2 is straightforward, because we only need Givens rotations to perform the QR decomposition of equation (22). This also means, that we don't have to perform an initialization when changing the constraints equations, because the PEs always work in orthogonal mode. The configurations of the processor array for both methods are given in Figure 4. Approach 1 (left hand

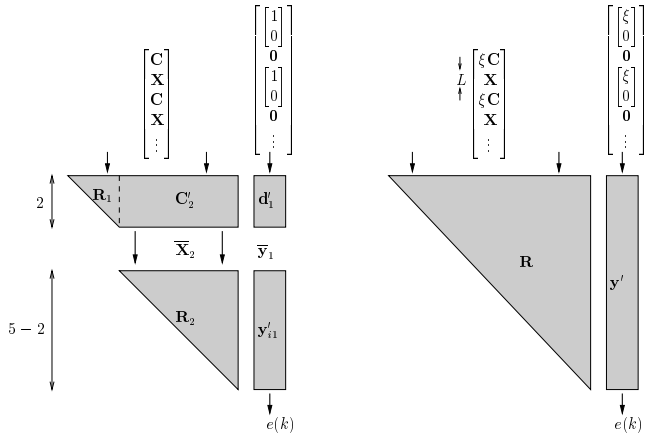

Figure 4: Processor Array

side) is advantageous in terms of a higher numerical stability, because there are no multiplications with $\xi \gg 1$ required. Furthermore we have to insert the constraints equations only if they have changed. In approach 2 (right hand side) we need to interweave the constraints matrix and the data matrix regularly, which results in a less efficient utilization of the processor array. On the other hand, if we change the constraints very fast, approach 2 is advantageous, because in this case no additional reconfiguration is needed, while approach 1 needs to perform an initialization step on every update.

5. SIMULATIONS AND DISCUSSION

The two different approaches are compared in BER simulations for different word lengths. The used scenario of a beamformer with five sensors is depicted in Figure 5 (left hand side). We assume 4-QAM modulated signals from three

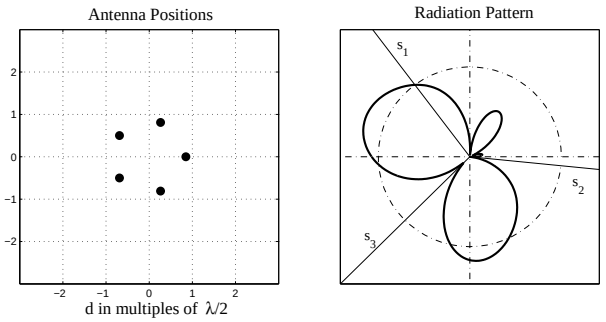

Figure 5: Used scenario for simulation

directions, where the directions of $s_1(t)$ and $s_2(t)$ are known. Signal $s_3(t)$ is an interfering signal from unknown direction. In Figure 5 (right hand side) a snapshot of the beamforming process for a desired signal $s_1(t)$ is depicted. It can be seen that both constraints ($|s_1(t)| = 1$ and $|s_2(t)| = 0$) are fulfilled, while the interfering signal $s_3(t)$ from unknown direction could not be completely eliminated. The BER results for this scenario using approach 1 are given in Figure 6. For the sake of comparison a lower bound is given by the BER of a transmission over a Gaussian channel without interfering signals $s_1(t)$ and $s_3(t)$. The optimum performance of the beamforming algorithm is defined by the floating point BER. As mentioned above, the processor elements are using fixed point arithmetics for computation. Obviously the implementation complexity can be reduced by decreasing the word length. We can see, that the curves for $w_d = 16$ bit and $w_d = 12$ bit match the floating point results. With $w_d = 8$ bit we get some loss in performance. The applicability for $w_d = 4$ bit is obviously not given. It seams feasible to use $w_d = 12$ or even $w_d = 8$ for a particular implementation.
The behavior of approach 2 under the same conditions is slightly different, as we can see in Figure 7. Here we already

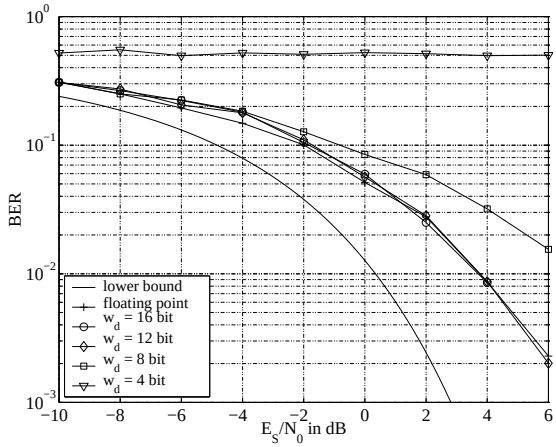

Figure 6: Simulation results for approach 1

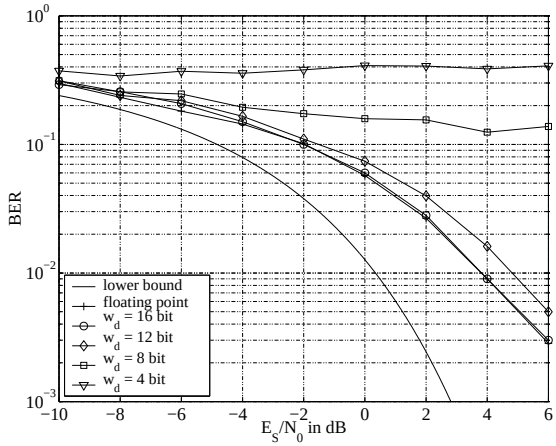

Figure 7: Simulation results for approach 2

see a performance loss with $w_d = 12$. In order to achieve floating point performance we need at least $w_d = 16$ bit. For a

particular implementation the word length required is higher compared to the implementation of approach 1.
Both approaches for solving the beamforming problem are generally applicable. Approach 1 has advantages in the numerical stability and the efficient utilization of the processor array. Furthermore the achievable BER with a particular word length is lower compared to approach 2. On the other hand approach 1 requires a mode change between linear and orthogonal activation when updating the constraints equations. This leads to a loss in data throughput due to the required configuration process.
Both approaches have to be seen as a trade-off between data throughput and BER performance. The choice of the right method depends on the particular application. For a given architecture method 1 might be used for lower rate applications with nearly floating point performance. If the focus is on high data throughput and a minor loss in BER performance can be accepted, method 2 might be the better choice.

6. CONCLUSION

In this paper we have presented the implementation of an MVDR beamformer on a reconfigurable hardware architecture using two different algorithms. While the first approach has better BER performance on an implementation with a particular word length, the second approach has advantages in the reconfiguration effort. Because it does not require any reconfiguration during the beamforming processing, approach 2 is better suited for high-speed applications. We have shown, that it is possible to use different methods to solve the particular linearly constrained least squares problem on the same platform. Dependent on the application the best compromise can be chosen and the respective configuration can be applied.
The use of a reconfigurable hardware architecture leads to a flexible platform for different processing tasks. Many applications, like equalization, detection and estimation can be mapped on this platform [12].

REFERENCES

[1] H. Bessai, *MIMO Signals and Systems*. Springer, Berlin, 2005.

[2] Walter H. W. Tuttlebee, *Software Defined Radio: Baseband Technology for Cellular Systems*. John Wiley and Sons Ltd, 2003.

[3] Jörg Brakensiek, Bernhard Oelkrug, Martin Bücker, "Software Radio Approach for Re-Configurable Multi-Standard Radios," in *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Lissabon, Portugal, 2002.

[4] R. Tessier and W. Burleson, "Reconfigurable Computing for Digital Signal Processing: A survey," *Journal of VLSI Signal Processing*, vol. 28, pp. 7–27, 2001.

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

[6] D.H. Johnson and D.E. Dudgeon, *Array Signal Processing*. Prentice-Hall, 1993.

[7] M. Otte, M. Bücker and J. Götze, "Complex CORDIC-like Algorithms for Linearly Constrained MVDR Beamforming," in *Proc. of IEEE International Zurich Seminar on Broadband Communication*, 2000.

[8] Sathish Chandran, *Advances in Direction-Of-Arrival Estimation*. Artech House Publishers, 2005.

[9] S. Haykin, *Adaptive Filter Theory,*. Prentice-Hall, 3rd Edition, 1996.

[10] V.N. Faddeeva, *Computational methods of linear algebra*. Dover books on advanced mathmatics. Dover Publ., New York, 1959.

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

[12] Marius Otte, *Matrixbasierte Signalverarbeitung auf rekonfigurierbaren CORDIC-Architekturen*. Shaker Verlag, 2004.