

# VLSI Architecture for Data-Reduced Steering Matrix Feedback in MIMO Systems

C. Studer, P. Luethi, and W. Fichtner

Integrated Systems Laboratory, ETH Zurich, Switzerland

email: {studer, luethi, fw}@iis.ee.ethz.ch

**Abstract**—Beamforming (BF) for multiple-input multiple-output (MIMO) wireless communications systems can improve the error rate performance by spatial separation of the transmitted data streams. BF requires to feed back steering matrices from the receiver to the transmitter. The usually large amount of feedback data asks for data reduction schemes. In this paper, we investigate the error rate performance/feedback rate trade-off associated with steering matrix data-reduction schemes and present a corresponding hardware-optimized compression/decompression architecture. Our VLSI implementation achieves up to 50% data reduction for  $4 \times 4$ -dimensional steering matrices without a significant decrease in terms of error rate performance at a circuit complexity of only 7k gate equivalents.

## I. INTRODUCTION

Multiple-input multiple-output (MIMO) technology offers increased spectral efficiency, compared to single-antenna systems, by transmitting multiple data streams concurrently and in the same frequency band [1]. Beamforming (BF) is considered in modern wireless standards, such as IEEE 802.11n [2], as a key technology to improve the error rate performance. BF requires to multiply the transmit vectors with a *steering matrix*. This matrix is obtained by computing the singular value decomposition (SVD) of the MIMO channel matrix. In practical systems, the VLSI implementations described in [3], [4] can be used to compute the SVD for BF.

In order to obtain the steering matrix in the transmitter, channel reciprocity or explicit feedback from the receiver to the transmitter can be used. In practice, exploiting reciprocity of the channel is often not possible due to strong RF impairments or if the up- and down-link are performed in different frequency bands. Hence, we focus on the scenario where the SVD is computed in the receiver and the steering matrices are fed back to the transmitter in a conventional data stream. A significant drawback of explicit steering matrix feedback is the potentially large amount of feedback data. In IEEE 802.11n, for example, up to 108 complex-valued  $4 \times 4$ -dimensional steering matrices need to be transmitted. To this end, algorithms that reduce the amount of feedback data (referred to as compression schemes) have been proposed in [5], [6].

*Contributions:* In this paper, we provide a bit-level comparison of three hardware-based data reduction schemes for explicit steering matrix feedback and discuss the resulting error rate performance/feedback rate trade-off for MIMO systems. For the most promising compression/decompression scheme, we describe a VLSI architecture that achieves up to

50% data reduction with near-optimal error rate performance. The final implementation is optimized in terms of hardware-efficiency, resulting in a high-throughput steering matrix compression/decompression unit that requires low silicon area.

*Outline:* The remainder of this paper is organized as follows. Sec. II introduces the MIMO system model with beamforming and describes three steering matrix feedback data reduction schemes. The corresponding compression/decompression architecture is described in Sec. III, hardware-efficiency optimizations and VLSI implementation results are given in Sec. IV, and we conclude in Sec. V.

## II. BEAMFORMING AND STEERING MATRIX FEEDBACK

Consider a MIMO system with  $M_T$  transmit and  $M_R$  receive antennas. The baseband-equivalent input-output relation corresponds to  $\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$ , where  $\mathbf{s}$  is the  $M_T$ -dimensional transmit vector,  $\mathbf{H}$  the  $M_R \times M_T$ -dimensional channel matrix,  $\mathbf{n}$  the  $M_R$ -dimensional additive Gaussian noise vector, and the  $M_R$ -dimensional receive vector is denoted by  $\mathbf{y}$ . One method to perform BF [7] is to transmit  $\tilde{\mathbf{s}} = \mathbf{V}\mathbf{s}$ , where  $\mathbf{V}$  corresponds to the  $M_T \times M_T$ -dimensional steering matrix. This matrix is obtained by computing the singular value decomposition (SVD) [8] of the channel matrix<sup>1</sup>

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \quad (1)$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are complex-valued unitary matrices and  $\mathbf{U}$  is of dimension  $M_R \times M_R$ . The real-valued  $M_R \times M_T$ -dimensional matrix  $\mathbf{\Sigma}$  contains  $r = \min\{M_R, M_T\}$  ordered singular values on its main diagonal. We assume that the receiver computes the steering matrix and explicitly feeds a data-reduced version back to the transmitter.

### A. Steering Matrix Quantization

A straightforward way to reduce the amount of steering matrix data is to quantize the real and imaginary parts of  $\mathbf{V}$ . Since steering matrices are unitary, all entries of  $\mathbf{V}$  satisfy  $\max\{|\Re(V_{i,j})|, |\Im(V_{i,j})|\} \leq 1$  ( $\forall i, j$ ) and hence, each complex-valued entry of  $\mathbf{V}$  can safely be quantized in two's-complement fixed-point format by using a sign bit and  $B_q - 1$  fraction bits for each real and imaginary part. This straightforward quantization scheme requires a total number of

$$S_v = B_q 2M_T^2 \quad (2)$$

bits per steering matrix. Note that the choice of  $B_q$  determines the error rate performance of BF (see Sec. II-D).

This work was supported by the STREP project No. IST-026905 (MASCOT) within the Sixth Framework Programme (FP6) of the European Commission.

<sup>1</sup>In the following, the superscripts  $T$  and  $H$  stand for transposition and conjugate transposition, respectively.

### B. Unique Steering Matrix Quantization

We emphasize that  $\mathbf{U}$  and the steering matrix  $\mathbf{V}$  in (1) are not unique. The application of a column-wise phase rotation applied to both unitary matrices with a  $M \times M$ -dimensional diagonal matrix

$$\mathbf{D}_l(\theta) = \text{diag}(\underbrace{1, \dots, 1}_{l-1}, e^{j\theta}, \underbrace{1, \dots, 1}_{M-l-2}) \quad M \in \{M_R, M_T\}$$

and by computing  $\tilde{\mathbf{U}} = \mathbf{U}\mathbf{D}_l(\theta)$  and  $\tilde{\mathbf{V}} = \mathbf{V}\mathbf{D}_l(\theta)$  for  $1 \geq l \geq M_T$ , ensures that  $\tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^H = \mathbf{U}\Sigma\mathbf{V}^H$  corresponds to another valid SVD of  $\mathbf{H}$ . Hence, an appropriate choice of phase rotations  $\mathbf{D}_l(\theta_l)$  for  $l = 1, 2, \dots, M_T$  can be exploited to obtain a *unique* steering matrix

$$\mathbf{V}_u = \mathbf{V} \prod_{l=1}^{M_T} \mathbf{D}_l(\theta_l) \quad (3)$$

such that the last row of  $\mathbf{V}_u$  only contains real-valued non-negative entries. Storage of the unique steering matrix only requires

$$S_u = B_q(2M_T^2 - M_T) - M_T \quad (4)$$

bits, since all imaginary parts in the last row are zero and the  $M_T$  sign bits of the corresponding real-valued entries can be neglected. We emphasize that (4) is lower than (2), especially for a small number of transmit antennas. Note however, the process of computing an unique steering matrix (3) requires dedicated signal processing hardware (see Sec. III).

### C. Steering Matrix Compression and Decompression

In order to further reduce the amount of data per steering matrix, more advanced data-reduction schemes (referred to as compression) have been proposed in [5], [6]. The key idea of steering matrix compression is to decompose a unitary matrix into a sequence of rotation angles required to perform a Givens-rotation-based QR decomposition of  $\mathbf{V}_u$ . The corresponding sequence of angles is obtained by choosing  $\phi_{kl}$  and  $\theta_{kl}$  such that<sup>2</sup>

$$\prod_{k=M_T-1}^1 \left( \prod_{l=k+1}^{M_T} \mathbf{G}_{kl}(\phi_{kl}) \prod_{l=k}^{M_T-1} \mathbf{D}_l(\theta_{kl}) \right) \mathbf{V}_u = \mathbf{I}_{M_T} \quad (5)$$

where  $\mathbf{I}_M$  is a  $M \times M$ -dimensional identity matrix and

$$\mathbf{G}_{kl}(\phi) = \begin{pmatrix} \mathbf{I}_{k-1} & 0 & 0 & 0 & 0 \\ 0 & \cos(\phi) & 0 & \sin(\phi) & 0 \\ 0 & 0 & \mathbf{I}_{l-k-1} & 0 & 0 \\ 0 & -\sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_{M_T-l} \end{pmatrix}$$

is the  $M_T \times M_T$ -dimensional Givens rotation matrix [8]. The angles  $\theta_{kl}$  in (5) are used to rotate complex-valued entries of  $\mathbf{V}_u$  to the real-axis and  $\phi_{kl}$  are used to zero out the lower-triangular part of the unique steering matrix.

Exact reconstruction (referred to as decompression) of  $\mathbf{V}_u$  is achieved by computing the inverse process of compression, using the same angles as obtained in (5). The unique steering

<sup>2</sup>Note that matrix-multiplication with the Pi-notation is defined from left to right, i.e.,  $\prod_{z=1}^3 \mathbf{X}_z = \mathbf{X}_1 \mathbf{X}_2 \mathbf{X}_3$ .

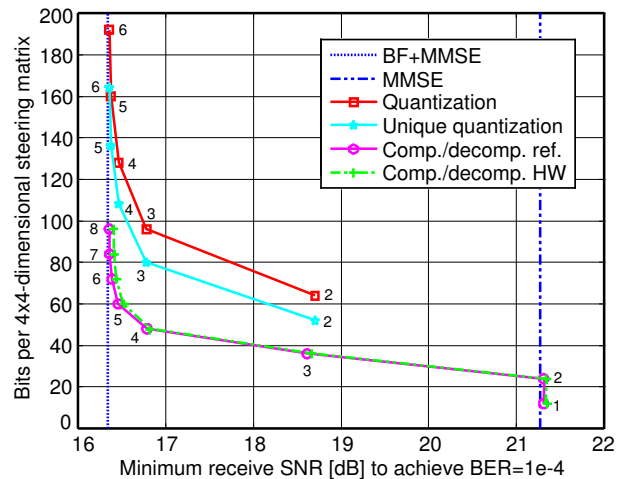


Fig. 1. Performance/feedback rate trade-off of steering matrix data-reduction schemes. The numbers next to the curves correspond to  $B_q$  for quantization-only schemes and  $B_a$  corresponds to steering matrix compression with the reference (floating-point) algorithm and to the hardware described in Sec. IV.

matrix is completely defined by  $(M_T - 1)M_T$  rotation angles  $\phi_{kl}$  and  $\theta_{kl}$  and requires

$$S_c = B_a(M_T - 1)M_T \quad (6)$$

bits for each compressed steering matrix.  $B_a$  corresponds to the number of bits per quantized angle. Since all angles are in the range  $[-\pi, \pi)$ , scaling by  $1/\pi$  converts the angular range to  $[-1, 1)$ , which allows a convenient quantization in signed two's complement format with a total number of  $B_a$  bits. For brevity of exposition, we consider equal quantization for  $\phi_{kl}$  and  $\theta_{kl}$  in the following.

### D. Performance/Feedback Rate Trade-Off

Compared to the quantization-only methods (2) and (4), steering matrix compression (6) can reduce the amount of bits per steering matrix by approximately  $B_a/(2B_q)$  for a large number of transmit antennas. The exact compression ratio is dependent on  $B_a$  and  $B_q$ , which have a strong impact on the error rate performance of the system. The performance impact of quantization and compression on the bit error rate (BER) is assessed by system simulations<sup>3</sup>. Fig. 1 shows the minimum SNR to achieve a target BER of  $10^{-4}$  dependent on the number of bits required per steering matrix.

The resulting trade-off between SNR operating point and feedback rate shows that quantization of the unique steering matrix  $\mathbf{V}_u$  reduces the storage requirements up to 25% compared to straightforward quantization of  $\mathbf{V}$ . Steering matrix compression further reduces the amount of feedback data by 25% to 50% depending on the SNR operating point. Only 6 bits per angle are sufficient to achieve near-optimal BER performance and requires 72 bits per  $4 \times 4$ -dimensional steering matrix. For the quantization-only schemes,  $B_q = 5$  attains near-optimal performance, resulting in 160 bits and 136 bits

<sup>3</sup>We consider a coded (rate 1/2 convolutional code with constraint length 7, generator polynomials [133<sub>o</sub> 171<sub>o</sub>], random interleaving) MIMO-OFDM system [7] with  $M_R = M_T = 4$ , 16-QAM (Gray mapping), 64 tones, and a linear soft-output MMSE detector. One code-block corresponds to 1024 bits, a TGN type C [9] channel model is used, and perfect channel state information at the transmitter and receiver is assumed.

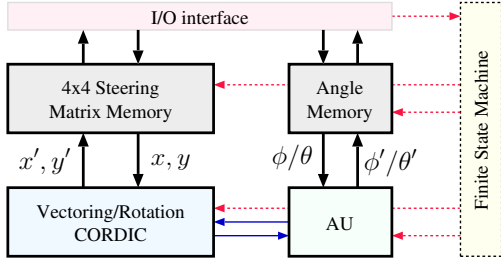


Fig. 2. Compression/decompression architecture for  $4 \times 4$  steering matrices.

per matrix for quantization of  $\mathbf{V}$  and quantization of  $\mathbf{V}_u$ , respectively. Note that setting  $B_q = 1$  results in a strong error floor since highly-quantized steering matrices are rank-reduced with high probability.

### III. VLSI ARCHITECTURE

To assess the signal processing overhead required to perform compression and decompression, a dedicated VLSI architecture is described in the following. In practice, steering matrix compression is only required in the receiver and decompression only in the transmitter. However, both tasks require the same amount of memory and a similar set of arithmetic operations. Thus, we present a *single* architecture that is capable to perform compression and decompression and is also able to compute the unique steering matrix.

The proposed architecture is depicted in Fig. 2 and contains a complex-valued  $4 \times 4$ -dimensional matrix memory and storage for 12 rotation angles. Latch arrays have been used to reduce the area of both memories, since the smallest available RAM macro cell was significantly larger. Further area reduction of the compression/decompression architecture is achieved by time-sharing of two coupled arithmetic units.

#### A. Vectoring/Rotation CORDIC with Enhanced Range

Steering matrix compression, decompression (5), and computing the unique matrix (3) requires two-dimensional rotations. CORDICs are a suitable tool to efficiently perform rotations in hardware [11] by decomposing two-dimensional rotations of a real-valued input vector  $\mathbf{v} = [x \ y]^T$  in  $R$  micro rotations according to

$$\mathbf{C}_i(d_i) = \kappa_i \begin{pmatrix} 1 & -d_i 2^{-(i-1)} \\ d_i 2^{-(i-1)} & 1 \end{pmatrix} \quad i = 1, 2, \dots, R$$

where  $\kappa_i = 1/\sqrt{1 + 2^{-2(i-1)}}$ . The total CORDIC rotation corresponds to  $\mathbf{v}' = \prod_{i=R}^1 \mathbf{C}_i(d_i) \mathbf{v}$ , which is determined by the sequence  $d_i \in \{-1, +1\}$  for  $i = 1, 2, \dots, R$ . Note that the architecture depicted in Fig. 3 only contains hardware-friendly arithmetic right shifts (ASRs), additions, subtractions, and two multiplications with the constant  $\kappa = \prod_{i=R}^1 \kappa_i$ . Unfortunately, the range of achievable rotation angles approximately corresponds to  $[-1.74, +1.74]$  radians and hence, the CORDIC needs to be modified in order to support all rotation angles.

*Vectoring Mode:* In this mode, the rotation sequence  $d_i$  is computed for  $i = 1, 2, \dots, R$ , such that for the input vector  $[x \ y]^T$ , the output corresponds to  $x' \approx \pm\sqrt{x^2 + y^2}$  and  $y' \approx 0$ . The required sequence can be extracted by choosing  $d_i = -\text{sign}(y_i) \text{sign}(x_i)$  for each step  $i$  and by

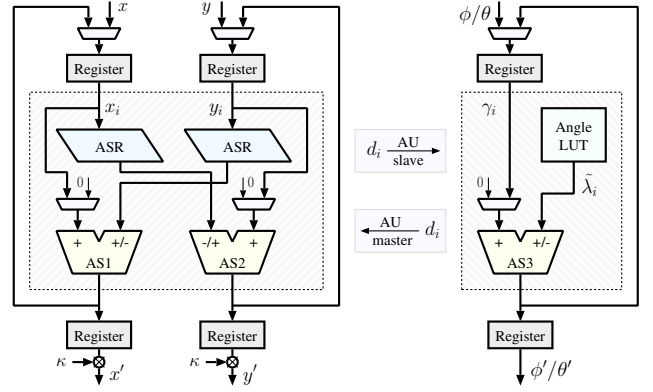


Fig. 3. Enhanced vectoring/rotation CORDIC on the left with the master/slave angular unit (AU) on the right. Multiple instantiation of the shaded boxes unroll the CORDIC/AU and can improve the hardware-efficiency (see Sec. IV).

setting the add/subtract mode in AS1 and AS2 (see Fig. 3) accordingly [10].

We emphasize that a real-valued *positive* output of  $x'$  in vectoring mode is essential for the computation of the unique steering matrix and for decompression (see Sec. III-B). To this end, the range of achievable rotation angles has been enhanced by performing an additional  $\pm\pi/2$  rad rotation

$$\mathbf{C}_0(d_0) = \begin{pmatrix} 0 & -d_0 \\ d_0 & 0 \end{pmatrix} \quad \text{with } d_0 \in \{-1, +1\}$$

prior to the first micro rotation of the CORDIC. Choosing  $d_0 = -\text{sign}(y_0)$  renders the output in vectoring mode non-negative, i.e.,  $x' \approx +\sqrt{x^2 + y^2}$  and  $y' \approx 0$ . We emphasize that this modification enhances the range of achievable rotation angles to approximately  $[-3.31, +3.31]$  radians at the cost of one additional micro-rotation step and two multiplexers at the input of AS1 and AS2 as shown in Fig. 3.

*Rotation Mode:* In this mode, the CORDIC in Fig. 3 is reused to perform a two-dimensional rotation of  $[x \ y]^T$  in the enhanced angular range, corresponding to a given  $d_i$  sequence for  $i = 0, 1, \dots, R$ .

#### B. Angular Master/Slave Unit

Steering matrix compression and decompression requires the extraction of rotation angles from Givens rotations and the rotation of vectors by the angles  $\theta_{kl}$  and  $\phi_{kl}$ , respectively. To perform both tasks, a master/slave angular unit (AU) has been designed (see Fig. 3) and is connected to the enhanced CORDIC architecture described in Sec. III-A.

*Slave Mode:* In this mode, the AU computes the rotation angle corresponding to the current rotation performed in the CORDIC in vectoring mode. Note that each micro rotation  $i$  of the enhanced CORDIC corresponds to the following angles

$$\lambda_i = \begin{cases} \pm\pi/2 & i = 0 \\ \pm \arctan(2^{-(i-1)}) & i = 1, 2, \dots, R. \end{cases} \quad (7)$$

Initializing  $\gamma_{-1} = 0$  and computing  $\gamma_i = \gamma_{i-1} - d_i \lambda_i$  for each micro rotation  $i$  yields the corresponding output angle  $\gamma_R$ . As depicted in Fig. 3, the computation of  $\gamma_R$  only requires an add/subtract unit (AS3) and a look-up table (Angle LUT) to store the  $R+1$  angles given in (7). Note that the angles in the LUT are scaled according to  $\tilde{\gamma}_i = \gamma_i/\pi$  such that the resulting angles are in the range  $[-1, 1]$ , which allows for convenient representation in two's complement fixed-point format.

TABLE I  
VLSI IMPLEMENTATION RESULTS OF THE STEERING MATRIX  
UNIQUIFY/COMPRESSION/DECOMPRESSION UNIT

Unroll Factor	1	2	3	6
Area <sup>a</sup> [kGE]	5.7	6.7	6.8	7.0
Max. clock freq. [MHz]	208	231	215	191
HW-efficiency <sup>b</sup> [kGE $\mu$ s]	9.8	5.7	4.5	3.1
Uniquify time [ $\mu$ s]	0.61	0.34	0.29	0.24
Comp. or Decomp. [ $\mu$ s]	1.73	0.84	0.64	0.44
Uniquify and Comp. [ $\mu$ s]	2.22	1.06	0.80	0.53

<sup>a</sup>One GE corresponds to the area of a two-input drive-one NAND gate.

<sup>b</sup>Hardware (HW) efficiency is measured in gate equivalents (GEs) times the time required to compress a  $4 \times 4$ -dimensional unique steering matrix.

*Master Mode:* The AU in master mode is only used in the steering matrix decompression phase. The purpose of this mode is to extract the corresponding micro-rotation sequence  $d_i$  ( $i = 0, 1, \dots, R$ ) for a given input angle, i.e., for either  $\theta$  or  $\phi$ . Simultaneously, the same rotation is performed in the CORDIC. Since the angles in the LUT (7) are stored in decreasing order, the micro-rotation sequence  $d_i$  can be derived by initializing  $\gamma_{-1}$  with either  $\theta$  or  $\phi$  and by computing

$$d_i = \text{sign}(\gamma_{i-1}) \quad \text{and} \quad \gamma_i = \gamma_{i-1} - d_i \lambda_i \quad (8)$$

in each micro-rotation  $i = 0, 1, \dots, R$ . We emphasize that computing (8) in the AU does not require any additional hardware (cf. Fig. 3) and is only possible due to enhancement of the CORDIC's angular range as described in Sec. III-A.

#### IV. IMPLEMENTATION RESULTS

The implementation results given in Tbl. I correspond to post-synthesis figures for  $0.18 \mu\text{m}$  (1P/6M) CMOS technology. All implementations are able to compute the unique steering matrix and perform compression and decompression of  $4 \times 4$ -dimensional steering matrices.

*Arithmetic Precision Optimization:* The fixed-point precision has been optimized to reduce circuit area and processing time. Simulations have shown that 8 bits per real and per imaginary part of  $\mathbf{V}$  and 7 bits to represent  $\theta_{kl}$  and  $\phi_{kl}$  are sufficient. The internal signals (cf. Fig. 3) of the CORDIC use 10 bits and the AU-internal signals  $\gamma_i$  and  $\lambda_i$  use 8 bits. Furthermore, six micro-rotation steps ( $R = 5$ ) are sufficient. The error rate performance of this hardware implementation is given in Fig. 1 and shows a tolerable performance loss compared to the floating-point reference algorithm.

*Impact of the Unroll Factor:* Further hardware-efficiency optimization is achieved by unrolling the CORDIC/AU. Tbl. I shows the final implementation results and illustrates the impact of the unroll factor (UF) to hardware-efficiency. No unrolling results in the smallest but least-efficient architecture. However, up to a threefold hardware-efficiency gain can be achieved by unrolling of the CORDIC/AU. The most efficient variant corresponds to that with an UF of six and achieves the fastest processing times among all other architectures.

*Comparison with Steering Matrix Computation:* A practical system that employs beamforming with data-reduced feedback needs to perform a SVD of the channel matrix in order to obtain the steering matrix. Tbl. II compares the effort (in terms of area and processing time) required to *compute* the steering matrix on MDU-I [3] and on the steering matrix

TABLE II  
COMPARISON OF STEERING MATRIX COMPUTATION UNITS WITH THE  
OPTIMIZED UF 6 UNIQUIFY/COMPRESSION/DECOMPRESSION UNIT

	MDU-I [3]	This Work	Total
Area [kGE]	42.3 (86%)	7.0 (14%)	49.3 (100%)
Time <sup>c</sup> [ $\mu$ s]	11.6 (96%)	0.53 (4%)	12.1 (100%)

<sup>c</sup>Corresponds to the SVD computation time using MDU-I [3] and to the uniquify and compression time using the UF 6 implementation of this work.

	SMCU [4]	This Work	Total
Area [kGE]	42.3 (86%)	7.0 (14%)	49.3 (100%)
Time <sup>d</sup> [ $\mu$ s]	3.3 (86%)	0.53 (14%)	3.83 (100%)

<sup>d</sup>Corresponds to the steering matrix computation time of the SMCU described in [4] and to the uniquify/compression time of our UF 6 architecture.

computation unit (SMCU) [4] with the efficiency-optimized (UF 6) steering matrix compression/decompression unit of this work. Compression only requires 14% of the total silicon area and 4% or 14% of the total computation time compared to the time required by the MDU-I or the SMCU, respectively. Thus, our compression/decompression architecture is a valuable addition to steering matrix computation units.

#### V. CONCLUSION

In this paper, we compared three different schemes for steering matrix data-reduction suitable for MIMO systems with beamforming. Investigation of the corresponding error rate performance/feedback rate trade-off have shown that sophisticated compression/decompression algorithms can achieve up to 50% data-reduction compared to straightforward quantization schemes. Furthermore, our VLSI implementation of steering matrix compression/decompression has proved to be suitable for MIMO systems with beamforming and performs all required signal-processing tasks in a hardware-efficient way with near-optimal error rate performance.

#### REFERENCES

- [1] H. Bölcskei, D. Gesbert, C. Papadias, and A. J. van der Veen, Eds., *Space-Time Wireless Systems: From Array Processing to MIMO Communications*. Cambridge Univ. Press, 2006.
- [2] IEEE Draft Specification, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Enhancements for higher throughput," *P802.11n/D1.0*, Mar. 2006.
- [3] C. Studer, P. Blösch, P. Friedli, and A. Burg, "Matrix decomposition architecture for MIMO systems: Design and implementation trade-offs," in *Proc. of the 41th Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2007.
- [4] C. Senning, C. Studer, P. Luethi, and W. Fichtner, "Hardware-efficient steering matrix computation architecture for MIMO communication systems," in *Proc. of the IEEE Int. Symp. on Circuits and Systems*, May 2008.
- [5] J. C. Roh and B. D. Rao, "Channel feedback quantization methods for MISO and MIMO systems," in *Proc. of IEEE PIMRC*, Sept. 2004, pp. 805–809.
- [6] M. A. Sadrabadi, A. K. Khandani, and F. Lahouti, "A new method of channel feedback quantization for high data rate MIMO systems," in *Proc. of the IEEE GLOBECOM '04*, vol. 1, Nov. 2004, pp. 91–95.
- [7] E. Akay, E. Sengul, and E. Ayanoglu, "Bit-interleaved coded multiple beamforming," *IEEE Trans. on Communications*, vol. 55, pp. 1805–1811, Sept. 2007.
- [8] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins Univ. Press, Baltimore and London, 1996.
- [9] V. Erceg et al., *TGn channel models*, May 2004, IEEE 802.11 document 03/940r4.
- [10] B. Parhami, *Computer Arithmetic Algorithms and Hardware Designs*. Oxford Univ. Press, New York, 2000.
- [11] J. R. Cavallaro and F. T. Luk, "CORDIC arithmetic for an SVD processor," *J. Parallel Distrib. Comput.*, vol. 5, no. 3, pp. 271–290, 1988.