A Low Complexity MIMO System Based On BLAST And Iterative Anti-Gray-Demapping

Alexander Boronka and Joachim Speidel
Institute of Telecommunications, University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Email: {boronka,speidel}@inue.uni-stuttgart.de

***Abstract*— We present a low complexity encoder and mapper for wireless multiple-input multiple-output (MIMO) systems. The BLAST (Bell Laboratories Layered Space-Time) algorithm is enhanced by a dedicated iterative QAM demapper with anti-Gray code. At the the transmitter a low complexity convolutional encoder and a QAM mapper with anti-Gray code are used, whose EXIT (Extrinsic Information Transfer) chart are matched to obtain a BER turbo cliff at low SNR. Computer simulations are presented for 4×4, 6×6, and 8×8 MIMO fading channels with additive white Gaussian noise. For a 6×6 MIMO channel our low complexity scheme can provide a turbo cliff at $E_b/N_0 = 8.5\,\mathrm{dB}$ for 16-QAM.**

I. Introduction

Telatar has shown in [1] mathematically that the capacity can be increased by the introduction of multiple-input multiple-output (MIMO) Gaussian channels. Foschini [2] and Foschini and Gans [3] proposed a new layered space-time architecture. In [4] and [5] this architecture is described in detail and named Bell Laboratories Layered Space-Time (BLAST). As an extension to BLAST turbo decoding [6] and space-time codes [7] have been presented. A simple and straightforward implementation of a coded BLAST transmitter is shown in Fig. 1. The binary output of the data source (information bits) $u(t_u)$ is encoded to yield $x'(t_x)$ (coded bits), interleaved resulting in $x(t_x)$, and mapped to $\mathbf{s}^{\mathrm{T}}(t_s)$ so that each of the M transmitters carries an independent data stream $s_i(t_s)$ where the index i denotes the number of the transmit antenna ($i = 1, \ldots, M$). t_u, t_x, and t_s are the discrete time instances of information bits, coded bits, and symbols, respectively, which are dropped in the following to simplify the notation. A random interleaver of size $S = 96000$ and a systematic convolutional code of rate $R_c = 1/2$, memory ν, feedforward polynomial G_{oct}, and feedback polynomial $G_{r,\mathrm{oct}}$ are used. The polynomials are usually given in octal numbers which is indicated by the index. This structure was called *vertical coding* [8].

A convenient way to describe a flat fading MIMO model is

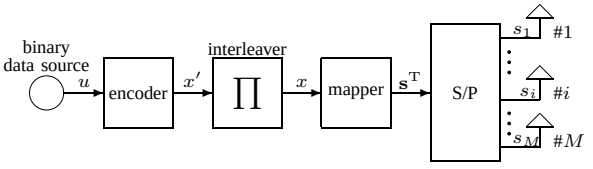

Fig. 1: Transmitter

the use of the matrix notation

$$\mathbf{r} = H\mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{s} = (s_1, \ldots, s_M)^{\mathrm{T}}$ is the symbol vector. Each component s_i is a complex QAM symbol to be sent by antenna i, ($i = 1, 2, \ldots, M$). $\mathbf{n} = (n_1, \ldots, n_N)^{\mathrm{T}}$ is an additive noise vector with components n_j, which are complex AWGN at receive antenna j, each with zero mean and variance $2\sigma_n^2$ ($j = 1, 2, \ldots, N$). This means that real and imaginary part of n_j are both Gaussian with variance $\sigma_n^2 = N_0/2$. N is the number of receive antennas. We consider $\mathbf{n}$ to be uncorrelated with expected value $\mathsf{E}\{\mathbf{n}\mathbf{n}^*\} = 2\sigma_n^2 I_N$. I_N is the identity matrix of size N. Furthermore, it is assumed that the vectors $\mathbf{s}$ and $\mathbf{n}$ are uncorrelated, i. e. $\mathsf{E}\{\mathbf{s}\mathbf{n}^*\} = 0$. $\mathbf{r} = (r_1, \ldots, r_N)^{\mathrm{T}}$ is the received symbol vector, where r_j is received by antenna j ($j = 1, 2, \ldots, N$). The channel matrix H is given by $H = (\mathbf{h}_1 \ldots \mathbf{h}_M)$ with $\mathbf{h}_i = (h_{1i}, \ldots, h_{Ni})^{\mathrm{T}}$ as a column vector of H. The impulse response h_{ji} from transmitter i to receiver j is modeled as a zero-mean, complex Gaussian random variable satisfying $\mathsf{E}\{|h_{ji}|^2\} = 1$ (i. e. the channel is passive). All entries of H are i.i.d. $(\cdot)^{\mathrm{T}}$ denotes the transpose and $(\cdot)^*$ the conjugate transpose. If not stated otherwise, bold characters are used for vectors in our text. (Complex) scalars are denoted by small and matrices by capital letters.

Assuming that we use the same mapping for all transmit antennas and mean energy per symbol E_s, the averaged total transmit power equals ME_s.

Using the column vectors of H we can write (1) as

$$\mathbf{r} = \sum_{i=1}^{M} \mathbf{h}_i s_i + \mathbf{n} \tag{2}$$

The BLAST system needs to have at least as much receive antennas N as transmit antennas M. While the transmitter does not need to have any knowledge about the channel, we assume that the receiver has full knowledge of H.

The paper is organized as follows. In Section II we describe the receiver and give short review of soft-output demapping of the BLAST symbols onto bits. Section III shows the method how we selected both, the constellation of the mapping and the encoder by means of the EXIT (extrinsic information transfer) chart. In Section IV simulation results are presented and discussed. Section V concludes this paper.

II. The Receiver

Fig. 2 shows the block diagram of the receiver. The original V-BLAST algorithm takes the N received samples r_j and performs the so called zero-forcing ordered successive interference cancellation (ZF-OSIC) or alternatively just the zero-forcing (ZF) algorithm. The latter operates without symbol cancellation [4]. However, we concentrate on an MMSE detector without interference cancellation (IC) which has a better performance than ZF [7]. Furthermore, omitting IC reduces the complexity of the receiver. The detection is based on equalization vectors $\mathbf{w}_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,N})$, $(i = 1, 2, \ldots, M)$ where $\mathbf{w}_i$ is the i^{th} row of the MMSE inverse of the channel matrix given by [9] $W = (\alpha I_M + H^* H)^{-1} H^*$ with $\alpha = (2\sigma_n^2)/E_s$ and $w_{i,j}$ is the j^{th} element of the i^{th} row of W. The detection of the transmitted symbols is done by

$$\mathbf{y} = W\mathbf{r} = W(H\mathbf{s} + \mathbf{n}) = WH\mathbf{s} + W\mathbf{n}$$

Introducing a so called *Post Detection Channel* (PDC) $\widetilde{H} = WH$ and a *Post Detection Noise* (PDN) $\widetilde{\mathbf{n}} = W\mathbf{n}$ results in

$$\mathbf{y} = \widetilde{H}\mathbf{s} + \widetilde{\mathbf{n}} = \sum_{i=1}^{M} \widetilde{\mathbf{h}}_i s_i + \widetilde{\mathbf{n}}$$

where $\widetilde{\mathbf{h}}_i$ is the i^{th} column of $\widetilde{H}$. Considering only one element of $\mathbf{y}$ —say y_{i_0}— we get

$$y_{i_0} = \underbrace{\widetilde{h}_{i_0,i_0} s_{i_0}}_{(A)} + \underbrace{\sum_{\substack{i=1 \\ i \neq i_0}}^{M} \widetilde{h}_{i_0,i} s_i}_{(B)} + \underbrace{\widetilde{n}_{i_0}}_{(C)}$$

where $\widetilde{h}_{j,i}$ is the j^{th} row and i^{th} column element of $\widetilde{H}$ and $\widetilde{n}_{i_0}$ is the $i_0{}^{\text{th}}$ element of $\widetilde{\mathbf{n}}$. Term (A) is the desired symbol s_{i_0} weighted with its PDC coefficient. (B) is the *Post Detection Inter Symbol Interference* (PD-ISI) which consists of all other symbols weighted with their corresponding PDC coefficients. Term (C) is the PDN for symbol s_{i_0}.

Note, that for ZF $\alpha = 0$ and $\widetilde{H} = I_M$ resulting in $(B) = 0$ and $\widetilde{h}_{i_0,i_0} = 1$.

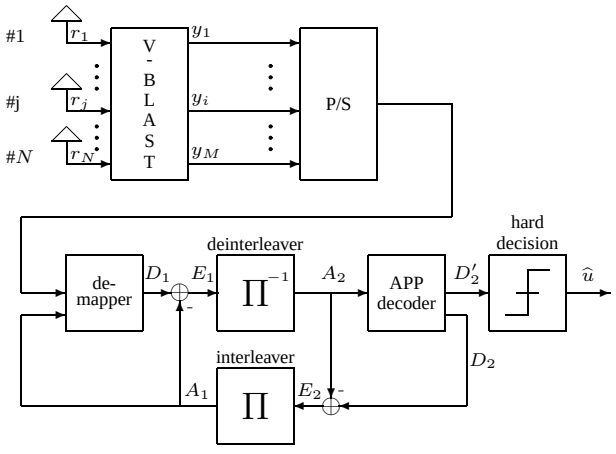

Fig. 2: Receiver structure

Since $\mathbf{n}$ is Gaussian (i. e. its entries have Gaussian pdf) $\widetilde{n}_{i_0}$ can be seen as a realization of a Gaussian process with variance $\widetilde{\sigma}_{n,i_0}^2 = \sigma_n^2 \sum_{i=1}^{N} |w_{i_0,i}|^2$ per direction. We can also compute the variance of the PD-ISI as $\sigma_{\text{PD-ISI},i_0}^2 = \sum_{\substack{i=1 \\ i \neq i_0}}^{M} |\widetilde{h}_{i_0,i}|^2 \mathsf{E}\{|s_i|^2\}/2$. This variance is updated within each iteration. The distribution of the PD-ISI is not Gaussian. However, to simplify the scheme, we summarize terms (B) and (C) into one single realization of a random process which we assume to be complex Gaussian with variance

$$\widetilde{\sigma}_{i_0}^2 = \widetilde{\sigma}_{n,i_0}^2 + \sigma_{\text{PD-ISI},i_0}^2$$

per direction. The subsequent MAP demapper in Fig. 2 performs bit-wise soft-output demapping described in [10]. Thus, the L-value of the k^{th} bit x_k of a symbol y_{i_0} is given by

$$L(x_k \mid y_{i_0}) = L_a(x_k) + \ln \frac{\sum_{\nu=0}^{2^{Q-1}-1} p\big(y_{i_0} | x_k = 1, x_{j,j=0,\ldots,Q-1,j\neq k}\big) \cdot \mathrm{e}^{\sum_{\substack{j=0 \\ j\neq k}}^{Q-1} x_j \cdot L_a(x_j)}}{\sum_{\nu=0}^{2^{Q-1}-1} p\big(y_{i_0} | x_k = 0, x_{j,j=0,\ldots,Q-1,j\neq k}\big) \cdot \mathrm{e}^{\sum_{\substack{j=0 \\ j\neq k}}^{Q-1} x_j \cdot L_a(x_j)}} \quad (3)$$

The values of the bits x_j in (3) accomplish the equation [10] $\sum_{j=0}^{k-1} x_j \cdot 2^j + \sum_{j=k+1}^{Q-1} x_j \cdot 2^{j-1} = \nu$. Equation (3) means that we set bit x_k to 1 in the numerator and to 0 in the denominator and permute over all other bits within the symbol. Q is the number of bits per symbol.

The BLAST algorithm provides no a priori information on the coded or information bits. So the complete a priori information for the very first pass is set to zero: $L_a(x_i) = 0 \ \forall i$.

Note that the MAP demapper is a simple 1x1 demapper as for systems with only 1 transmitter and 1 receiver. Thus, its complexity is independent of the number of transmit or receive antennas and depends only on the number of bits per symbol.

The further iterative processing in Fig. 2 is well known: The a priori information A_1 is subtracted from the a posteriori information D_1 given by the L-values in (3). After deinterleaving of E_1, the result A_2 is passed on as new a priori information to the soft input soft output APP decoder. The decoder calculates the new a posteriori information D_2. The extrinsic information E_2 is interleaved resulting in new a priori information A_1 for the demapper. After some iterations the output L-values D_2' of the decoder are hard decided to obtain estimates $\widehat{u}$ on the information bits. In the following we refer to the information of A_i and E_i as I_{A_i} and I_{E_i}, respectively $(i = 1, 2)$. Partly, we will omit index i for simplification.

III. Mapper, Decoder, and EXIT chart

In the design of the presented transmitter and receiver structure, we have two degrees of freedom besides the choice of the number of transmit and receive antennas. First, if we keep the number of bits per symbol fixed, we can determine the signal constellation, e. g. QAM. In particular we can choose how the bits are mapped to the symbols. Second, the

encoder/decoder has to be designed to achieve the desired performance. A leading and practical method how to design both, mapper and encoder, is given by the EXIT chart technique which is presented in detail by S. ten Brink in [11], [12]. In the following we present a two step method. First, a mapping is chosen independent of the encoder. After that, a proper encoder/decoder is selected to match with the transfer characteristics of the mapper. As is known from [11], [12], the encoder can be selected to obtain an early 'turbo cliff' at low SNR, or a low error floor, or a minimum number of iterations. We focus on a design for an early turbo cliff. Similar as shown in [10], Gray mapping combined with iterative demapping and decoding does not yield a significant improvement. Thus, it is absolutely necessary to take anti-Gray mappings into consideration in the following.

A. QPSK

For QPSK (or 4-QAM) there are in principal $4! = 24$ different possibilities how to map the 4 different bit patterns 00, 01, 10, and 11 to the 4 constellation points. However, due to symmetry with respect to axes, rotary symmetry, interchange of 0 and 1, and flipping the bit order, it turns out that finally only 2 mappings with significantly different EXIT charts remain: Gray mapping and the anti-Gray mapping. Thus, the degrees of freedom are very limited. This is the reason, why we focus in this paper on 16-QAM.

B. 16-QAM

For a 16-QAM there are in principal $16! \approx 2 \cdot 10^{13}$ different possible mappings. Also in this case there are a lot of symmetries among the mappings with respect to their transfer characteristics in the EXIT chart. However, still enough different mappings remain. To find the best mapping, we have to review briefly some fundamentals of EXIT chart technique. The ultimate goal of the decoder is to minimize the bit error ratio (BER). As is well known from [11], [12], BER of an iterative decoding scheme is restricted by the intersection point of the EXIT characteristics of the inner and outer decoder which lies closest to the origin ($I_A = 0, I_E = 0$) of the chart. Here, we consider the 16-QAM mapper as 'inner encoder' with code rate 1. To achieve low BER the intersection point should be placed as far as possible away from the origin of the EXIT chart. The intersection point depends on the signal to noise ratio (SNR) because the transfer characteristic of the mapper depends on the SNR. Leaving the SNR constant, the intersection point differs from mapping to mapping (excluding symmetric mappings). We observe $I_E(I_A = 1) < 1$ for any mapping. In addition, the transfer characteristic of the outer decoder is independent of the SNR. This leads us to the following search criterion: Find the mapping, for which

$$\max_{\mathcal{Q}} \{I_E(I_A = 1) \mid \mathrm{SNR} = \mathrm{const.}\} \tag{4}$$

holds, where $\mathcal{Q}$ is the set of all possible mappings. As an example we consider a MIMO scheme with $M = N = 6$. Unfortunately, there is no simple algorithm to evaluate (4). Therefore we have executed a random search over 1,500 different mappings and found the mapping depicted in Fig. 3. The numbers in Fig. 3 represent the bit patterns in decimal notation. In addition, we have examined some other mappings described in Section IV. However, none of them reached the performance of the random searched one. We introduce the notation of the mapping in Fig. 3 as (read from bottom left to top right)

$$\mathrm{map}_1 = [11, 2, 12, 1, 7, 9, 6, 15, 4, 10, 3, 5, 14, 13, 0, 8].$$

The mapping in Fig. 3 has no noticeable structure with respect to the bit patterns and is of anti-Gray type.

C. Encoder and Decoder

After the mapping is defined the task is to find an encoder/decoder transfer characteristic which matches optimally to the mapping. In the EXIT chart the transfer characteristic of a simple outer (memoryless) repetition code is the diagonal line, whereas the transfer characteristic of a serially concatenated code with infinite memory is a horizontal line $I_E(I_A) = \mathrm{const.}$ [11]. Therefore, the number of memory elements may serve as the first decision criterion for selection of a proper encoder and corresponding decoder. We restrict ourselves to systematic convolutional encoders of rate $R_c = 1/2$. The encoders may or may not be recursive. By extensive search of encoders up to memory 4, we have found that in principle a code with memory 2 results in a good match to the mapping defined in subsection III-B. While the gradient of the transfer characteristic is mainly defined by the memory of the encoder, the generator polynomial has marginal impact and is still subject to optimization. The general structure of a memory 2 recursive systematic code is shown in Fig. 4 ($g_i, g_{\mathrm{r},i} \in \{0, 1\}, i = 0, 1, 2$). Note that $g_{\mathrm{r},0} = 1$ by convention. Since at least one of the feedforward coefficients g_0, g_1, or g_2 has to be nonzero, and at least g_2 or $g_{\mathrm{r},2}$ has to be nonzero, there are 22 different possible generator polynomials. A Comparison of those 22 encoders shows that only 3 have different transfer characteristics. These are

$$\begin{aligned} \text{encoder I:} \quad & G_{\mathrm{oct}} = 7, G_{r,\mathrm{oct}} = 4 \\ \text{encoder II:} \quad & G_{\mathrm{oct}} = 3, G_{r,\mathrm{oct}} = 7 \\ \text{encoder III:} \quad & G_{\mathrm{oct}} = 5, G_{r,\mathrm{oct}} = 7 \end{aligned}$$

Note that encoder I is non-recursive since $g_{\mathrm{r},1} = g_{\mathrm{r},2} = 0$, and encoders II and III are recursive. Interestingly, the transfer

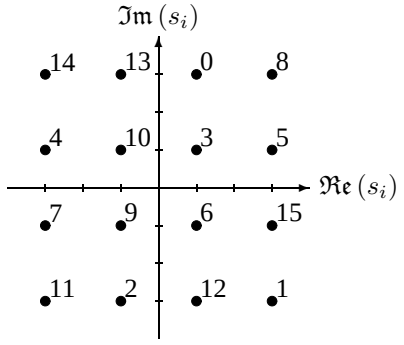

Fig. 3: 16-QAM to maximize (4)

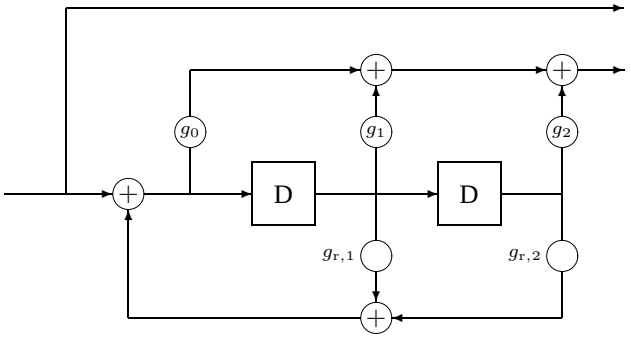

Fig. 4: General structure of a memory 2 RSC

characteristic of encoder I matches best to our mapping.

D. EXIT-Chart

Fig. 5 shows the simulated transfer characteristics of the mapper (indicated by index 1) and the outer decoder corresponding to encoder I (indicated by index 2) for $E_b/N_0 = 8.5\,\mathrm{dB}$. For the outer decoder the axes are swapped. The definition of E_b/N_0 is given in Section IV. The transfer characteristic of the mapper starts at $I_{E_1}(I_{A_1} = 0) \approx 0.26$ and ends at $I_{E_1}(I_{A_1} = 1) \approx 0.86$. The decoder characteristic starts at the origin $I_{E_2}(I_{A_2} = 0) = 0$ and ends at $I_{E_2}(I_{A_2} = 1) = 1$. The point of intersection of the two curves is at about $(I_{A_1}, I_{E_1}) = (0.98, 0.85)$. This is sufficient to reach a very low error floor. From the EXIT chart it can also be seen that over a wide range the two curves are almost parallel.

IV. Simulation Results

In the following we present simulation results of the BER for different scenarios. The BER is plotted over E_b/N_0 with $E_b/N_0 = \frac{E_s \cdot N}{R_c \cdot Q \cdot 2\sigma_n^2}$ where $\frac{E_s \cdot N}{R_c \cdot Q}$ is the transmitted energy per information bit at the receiver. We compare the mapping in Fig. 3 with two hand-designed mappings:

$$\mathrm{map}_{\mathrm{Gray}} = [3, 1, 9, 11, 2, 0, 8, 10, 6, 4, 12, 14, 7, 5, 13, 15]$$
$$\mathrm{map}_2 = [1, 14, 8, 15, 4, 3, 13, 10, 9, 0, 6, 5, 7, 12, 11, 2]\ .$$

$\mathrm{map}_{\mathrm{Gray}}$ is Gray mapping with the well-known property that two adjacent symbols differ only in one bit whereas with mapping map_2 every pair of two adjacent symbols differ in at least two bit. The results are shown in Fig. 6 for $M = N = 6$. The BER chart shows that the turbo cliff for map_1 occurs at about 8.4 dB. This is in line with what we expected from the EXIT chart. Gray-mapping is outperformed after few iterations. We observed a small reduction of BER when using Gray-mapping only in the first iteration. Further iterations did not yield significant lower BER. The error floor is decreasing with increasing SNR and map_1 reaches a BER of $2 \cdot 10^{-4}$ at 10 dB after 9 iterations (not shown). The reason for the floor is that the transfer characteristic of the mapper does not reach the upper right corner of the EXIT chart. Thus the intersection point does not reach the upper right corner. However, the BER is sufficiently low for a wide range of applications using an additional outer code with code rate < 1.

map_2, which was designed by hand, has a similar performance as the map_1. The difference between the turbo cliffs is small. But the error floor is significantly higher. This is due to the fact, that for map_2 $I_E(I_A = 1) \approx 0.837$ is lower than the corresponding value 0.86 of map_1. Remember that the presented combination of mapper and encoder was designed to achieve low BER for low SNR rather than reaching low BER within only a few iterations. With our design we achieved an early turbo cliff and a reasonable low error floor. During our investigation we found several other combinations of mapper and encoder to fulfill one or two of the criteria mentioned above. As a result one could say: It is relatively easy to achieve a low error floor or a small number of iterations simply by increasing the memory of the encoder (which automatically shifts the turbo cliff to higher SNR values). But it is much more complicated to shift the turbo cliff to low SNR without increasing the error floor.

In Fig. 7 mapper and encoder are the same as in Fig. 6 but the number of transmit and receive antennas is varied. As can be seen, reducing the number of antennas from 6 to 4 (which automatically reduced the bit rate because the constellation size is the same) results in an earlier turbo cliff but the floor is almost the same. Increasing the number of antennas from 6 to 8 also results in almost the same error floor, but as expected,

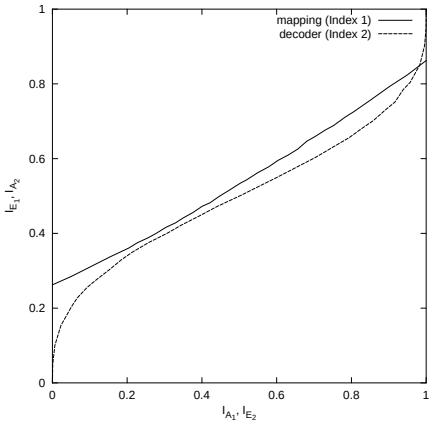

Fig. 5: EXIT chart of map_1 for $E_b/N_0 = 8.5\,\mathrm{dB}$ and decoder I

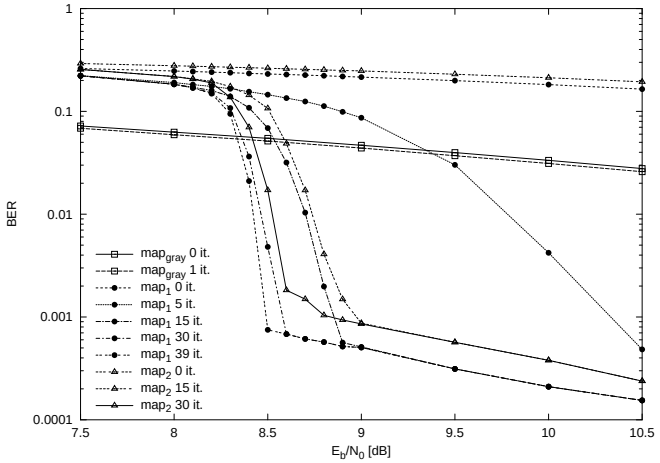

Fig. 6: BER for different mappings with outer encoder I and $M = N = 6$

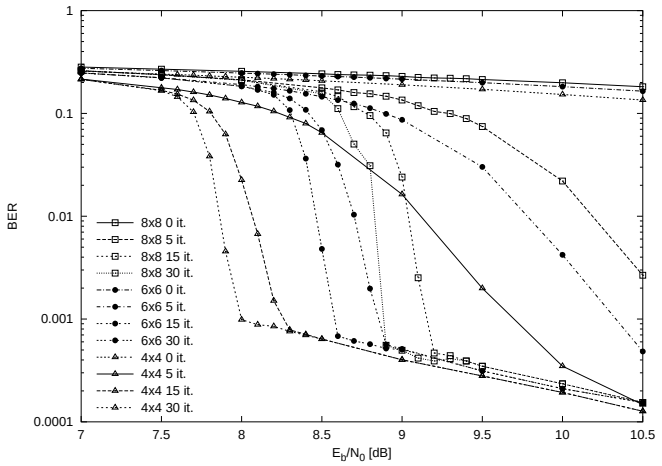

Fig. 7: BER for different numbers of antennas using map_1 and encoder I

in a later turbo cliff. However, the distance between the turbo cliffs is smaller when increasing the number of antennas from 6 to 8 than from 4 to 6.

A. Some remarks on the complexity

In this subsection we summarize the main points which determine the complexity of the system. The BLAST detector has to compute only one single MMSE inverse per symbol vector. Therefore, $\alpha = (2\sigma_n^2)/E_s$ has to be estimated at the receiver. The complexity of the MAP demapper increases only with the number of bit per symbol and not with the number of transmitters or receivers. The demapper is applied M times per symbol vector per iteration. The encoder is a simple non-recursive systematic encoder with only memory 2. Thus, the system complexity is much lower than that of an optimal MAP demapper which is searching over all possible vectors $\mathbf{s}$. In particular taking our example of 16-QAM with $Q = 4$ bit per symbol and $M = 6$ transmitters, the complexity per iteration of an MAP demapper is $2^{Q \cdot M} = 2^{24} \approx 1.67 \cdot 10^7$ compared to $M \cdot 2^Q = 6 \cdot 2^4 = 96$ of our scheme. Note, that we only apply a simple 1x1 demapper M times per iteration. As shown in [13] the BER floor can be reduced by MAP demapping to the expense of a higher receiver complexity. The complexity of our scheme is also lower compared to solutions, in which multiple inverse matrices have to be computed.

V. Conclusion

We have presented an enhanced wireless MIMO system with BLAST detection. At the transmitter a convolutional encoder is used as outer encoder. The QAM mapper is considered as a serially concatenated inner encoder with code rate 1. At the receiver the V-BLAST detection algorithm is enhanced by an iterative QAM demapper and APP decoder. Convolutional encoding and mapping are designed in such a way that BER as a function of SNR exhibits an early turbo cliff, i. e. a steep decent at low SNR. To reach this goal, the EXIT chart technique was employed. Out of the great variety of convolutional encoders up to memory 4 and the vast of about 10^{13} different mappings for 16-QAM the best match was selected using their EXIT charts (Fig. 5). As a result, a non-recursive convolutional encoder with memory 2 and a dedicated anti-Gray mapping turned out to be the best solution. Our simulations have also shown, that the impact on the BER turbo cliff of the generator polynomials of the convolutional encoder (both recursive and non-recursive) is quite small. With the presented method, we have investigated BER for 4×4, 6×6, and 8×8 MIMO fading channels with AWGN (Figs. 6 and 7). For the 6×6 MIMO channel a turbo cliff at $E_b/N_0 = 8.5\,\mathrm{dB}$ was achieved, which is a remarkable result for such a low complexity scheme. The BER floor of about 0.001 can be further reduced by adding a simple outer encoder and decoder to the system. The complexity of our algorithm is only proportional to $M \cdot 2^Q$ compared to $2^{Q \cdot M}$ for the existing MAP detector without BLAST, which is obviously much more complex.

References

[1] İ. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Bell Labs Internal Technical Memo*, June/October 1995, Reprint: *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585-595, Nov./Dec. 1999. [Online]. Available: http://mars.bell-labs.com

[2] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, Autumn 1996.

[3] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–335, 1998.

[4] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *1998 International Symposium on Signals, Systems, and Electronics ISSSE'98*, Pisa, Italy, September 30, 1998.

[5] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," *ELECTRONIC LETTERS*, vol. 35, no. 1, pp. 14–15, January 7, 1999.

[6] G. Bauch and J. Hagenauer, "Multiple antenna systems: Capacities, transmit diversity and turbo processing," in *4th International ITG Conference on Source and Channel Coding*, January 28-30, 2002, pp. 387–398.

[7] S. Bäro, G. Bauch, A. Pavlic, and A. Semmler, "Improving BLAST performance using space-time block codes and turbo decoding," in *IEEE Global Telecommunications Conference 2000 (Globecom)*, November 2000, pp. 1067–1071.

[8] X. Li, H. Huang, G. J. Foschini, and R. A. Valenzuela, "Effects of iterative detection and decoding on the performance of BLAST," in *IEEE Global Telecommunications Conference 2000 (Globecom)*, San Francisco, USA, November 27 - December 1, 2000, pp. 1061–1066.

[9] A. van Zelst, "Extending the capacity of next generation wireless LANs using space division multiplexing combined with OFDM," Master's thesis, Technische Universiteit Eindhoven, Faculty of Electrical Engineering, Section of Telecommunication Technology and Electromagnetics, Netherlands, 1999.

[10] S. ten Brink, J. Speidel, and R.-H. Yan, "Iterative demapping and decoding for multilevel modulation," in *IEEE Global Telecommunications Conference 1998 (Globecom)*, November 1998, pp. 579–584.

[11] S. ten Brink, "Design of concatenated coding schemes based on iterative decoding convergence," Dr.-Ing. dissertation (Ph.D.), University of Stuttgart, 2001, published at Shaker Aachen, Germany, ISBN 3-8322-0684-1.

[12] ——, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, October 2001.

[13] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communciations*, vol. 51, no. 3, pp. 389–399, March 2003.